

Investigation on Statistical-based Detection Techniques For  
Control Valve Stiction Detection

by

Khadijah binti Abdullah

15250

Dissertation submitted in partial fulfillment of  
the requirements for the  
Bachelor of Engineering (Hons)  
Chemical Engineering

SEPTEMBER 2015

Universiti Teknologi PETRONAS  
32610, Bandar Seri Iskandar,  
Perak

CERTIFICATION OF APPROVAL

**Investigation on Statistical-based Detection Techniques For  
Control Valve Stiction Detection**

by

Khadijah binti Abdullah

15250

A project dissertation submitted to the  
Chemical Engineering Programme  
Universiti Teknologi PETRONAS  
in partial fulfilment of the requirement for the  
BACHELOR OF ENGINEERING (Hons)  
(CHEMICAL ENGINEERING)

Approved by,

---

(Dr Haslinda binti Zabiri)

UNIVERSITI TEKNOLOGI PETRONAS  
BANDAR SERI ISKANDAR, PERAK  
September 2015

## CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgment, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

---

KHADIJAH BINTI ABDULLAH

## **ABSTRACT**

Control valve stiction is one of the main causes that can affect the performance of a control loop. As the final control element, it can cause disruptions towards the operations especially on the plant production of oil and gas industry. An initiative had been made in 1989 where a stiction detection method is first developed to detect stiction in control valve. Since then, many methods are produced and redeveloped but only few uses the statistical-based methods. Hence, this project will cover statistical-based methods which had once been used for fault detection and will be tested for the effectiveness in detecting control valve stiction. Two case studies are chosen which are simulation case study involving stiction and non-stiction conditions namely, well-tuned controller (Base case), tightly-tuned controller (Case 1), presence of disturbances controller (Case 2) and presence of stiction controller (Case 3). Another case study is the real industrial data from a chemical plant. The process output (pv) and controller output (op) for each case are generated based on nonlinear principle component analysis (NLPCA) method. Two statistical-based methods are chosen to be tested which are generalized likelihood ratio (GLR) test as well as error testing method. Based on the results of testing for method 1, there are some limitations for GLR test method to detect between the presence of stiction and the presence of disturbances in the system. However, for method 2, it can be seen that the error testing method focusing at the root mean squared error (RMSE) calculation is an effective tool and method to detect stiction and manage to differentiate stiction and non-stiction system for the simulation and real industrial case studies chosen.

## **ACKNOWLEDGEMENT**

First and Foremost, the author would like to humbly praise Allah the Almighty for His blessings upon the completion of this Final Year Project (FYP). The success of this project requires a lot of guidance from several parties.

The author would like to give gratitude to the supervisor, Dr Haslinda binti Zabiri for her support and guidance throughout the project completion. Thank you for the fruitful knowledge, advices and guidance. The appreciation also goes to Miss Nur Amalina Shairah, for her assistance in helping the author throughout the semester. Not to forget, thank you to lecturers and colleagues for all the encouragement from the beginning until the end.

Last but not least, to the authors' parents, Abdullah bin Hassan and Haspawiza binti Ibrahim who are very supportive. Their encouragements motivate and strengthen the author throughout this long period. Thank you all.

## TABLE OF CONTENTS

<b>CERTIFICATION OF APPROVAL .....</b>	<b>ii</b>
<b>CERTIFICATION OF ORIGINALITY .....</b>	<b>iii</b>
<b>ABSTRACT .....</b>	<b>iv</b>
<b>ACKNOWLEDGEMENT .....</b>	<b>v</b>
<b>TABLE OF CONTENTS .....</b>	<b>vi</b>
<b>LIST OF FIGURES .....</b>	<b>viii</b>
<b>LIST OF TABLES.....</b>	<b>ix</b>
<b>CHAPTER 1 :        INTRODUCTION .....</b>	<b>1</b>
1.1 Background .....	1
1.2 Problem Statement .....	2
1.3 Objectives and Scope of Study.....	3
<b>CHAPTER 2        LITERATURE REVIEW .....</b>	<b>4</b>
2.1 Introduction .....	4
2.2 Control Valve .....	4
2.3 Control Valve Stiction.....	5
2.4 Statistical-based Methods for Fault Detection .....	9
<b>CHAPTER 3 :        METHODOLOGY .....</b>	<b>12</b>
3.1 Introduction .....	12
3.2 General Project Work.....	12
3.2.1 Development of pv and op data.....	13
3.2.1.1 Simulation case study .....	13
3.2.1.2. Industrial case study.....	15
3.2.2 Statistical-based methods steps .....	16
3.2.2.1 Generalized likelihood ratio (GLR) test .....	16
3.2.2.2 Error Testing Method.....	17

	3.3 Project Gannt Chart and Key Milestone .....	20
	3.4 Tools.....	21
<b>CHAPTER 4 :</b>	<b>RESULTS AND DISCUSSION.....</b>	<b>22</b>
	4.1 Introduction .....	22
	4.2 Generalized Likelihood Ratio (GLR) Test.....	22
	4.3 Error Testing Method Test .....	27
<b>CHAPTER 5 :</b>	<b>CONCLUSIONS AND RECOMMENDATION .....</b>	<b>37</b>
	5.1 Conclusion.....	37
	5.2 Recommendation.....	37
<b>REFERENCES.....</b>		<b>38</b>
<b>APPENDICES.....</b>		<b>41</b>

## LIST OF FIGURES

Figure 1.1	Common problems and causes of poor control performance [3]	2
Figure 2.1	Cross-sectional diagram of control valve [7]	4
Figure 2.2	Typical closed loop control with sticky valve	6
Figure 2.3	Typical behavior of a sticky valve [6]	6
Figure 2.4	Comparison of stiction detection methods	8
Figure 2.5	Classifications Scheme for Fault Detection [20]	9
Figure 3.1	Process flow chart	13
Figure 3.2	Block diagram for well-tuned controller	14
Figure 3.3	Block diagram for excessive integral action controller	14
Figure 3.4	Block diagram for external oscillation controller	15
Figure 3.5	Block diagram for control system with presence of stiction	15
Figure 3.6	Flowchart of GLR testing method	17
Figure 3.7	Curve Fitting Toolbox in Matlab	18
Figure 3.8	Flowchart of error testing method	19
Figure 4.1	The pv-op plotted graph for base case	23
Figure 4.2	The pv-op plotted graph for case 1	24
Figure 4.3	The pv-op plotted graph for case 2	24
Figure 4.4	The pv-op plotted graph for case 3	25
Figure 4.5	Polynomial plot for base case	28
Figure 4.6	Polynomial plot for case 1	28
Figure 4.7	Polynomial plot for case 2	29
Figure 4.8	Polynomial plot for case 3	29
Figure 4.9	Comparison on the regression values	30
Figure 4.10	Comparison on the sum squared of error values	31
Figure 4.11	Comparison on the RMSE for all cases	32
Figure 4.12	Polynomial plot for loop 1	33
Figure 4.13	Polynomial plot for loop 2	34
Figure 4.14	Polynomial plot for loop 3	34
Figure 4.15	Polynomial plot for loop 4	34
Figure 4.16	Polynomial plot for loop 5	35



## LIST OF TABLES

Table 2.1	Four general types of valves [8]	5
Table 2.2	Stiction detection methods with descriptions	7
Table 2.3	Fault detection methods using statistical-based method	10
Table 3.1	Industrial data from flow control loops	16
Table 3.2	Project gannt chart and key milestone	20
Table 4.1	The chosen pv and op data for lowest regression values	25
Table 4.2	Summary of results generated for all case studies	26
Table 4.3	Slope calculations for simulation case study	27
Table 4.4	Summary of error calculation for each case	30
Table 4.5	Slope calculations for real industrial data	32
Table 4.6	Error calculation for each industrial case study	35
Table 4.7	Comparison of the proposed method with fuzzy clustering method on the industrial data set	36

# **CHAPTER 1**

## **INTRODUCTION**

This chapter provides some background knowledge related to the field of study. The subsections divided include the background and relation of stiction with control loop performances, problem statement and the objectives to be achieved by the end of this project. The scope of study also covers the methods chosen for this project.

### **1.1 Background**

For a normal operating of a chemical plant, hundreds of control loops are involved in controlling the process to get the best quality products for customers. Control valve is the final control element for every process loop. In order to optimize and maximize productivity, the control valves should always be in their best performance. However, according to Paulonis and Cox (2003), around 20-30% of the control loops are suffering with the nonlinearities problem which is valve stiction [1]. These have caused oscillatory results to the process output. Figure 1.1 shows the common problems for poor control performances which includes stiction.

Stiction is derived from the word “static” and “friction”. According to the Instrument Society of America (ISA,1979), “Stiction is the resistance to start of motion, usually measured as the difference between the driving values required to overcome static friction upscale and downscale”[2]. There are abundant methods that have been developed in order to detect the stiction of control valve. However, this project paper will focus only to the statistical-based methods.

Problem	Possible causes
Oscillations	Valve/damper stiction
	Controller tuning
	Cyclical interaction
	Deadband
	Process issues
Sluggish response	Controller tuning
	Excessive filtering
	Improper disturbance handling
	Deadband
Noise	Incorrect measurement technology
	Improper installation of measurement device
	Turbulence or volatility
	Insufficient filtering

Figure 1.1 Common problems and causes of poor control performance [3]

## 1.2 Problem Statement

Control valve plays an important role as the final control element to help in the production of products at good condition. However, this device has to undergo maintenance process from time to time to check its performance based on the process output. Throughout the operation, a nonlinearity process output may occur and one of the reasons is due to the control valve stiction. This will hinder the process from running smoothly. Therefore, the detection of this problem in the control loop is very important to make any changes and correction at the control valve.

Since the early stage of stiction detection, there are many types of approaches that have been made. However, in the present work, only few of the methods used the statistical-based techniques to detect valve stiction. Meanwhile, there are a lot of statistical-based methods used for fault detection but only few are applied to detect valve stiction.

### **1.3 Objectives and Scope of Study**

The main objectives of this project are:

- 1) To investigate on the existing statistical-based methods for efficient fault detection
- 2) To simulate the effectiveness of the method on detecting control valve stiction

The scope of study for this project will be covering the statistical-based methods on detecting the control valve stiction. The case studies of the feedback control system for simulation and real industrial data had been developed in Matlab software using nonlinear principle component analysis (NLPCA) method by H. Zabiri and Ramasamy [4]. This method managed to capture the stiction behavior based on the curve shape formed. However, the detection of stiction will be improved with the generation of controller output (op) and process output (pv) data that are further tested with statistical-based methods namely Generalized Likelihood Ratio (GLR) and Error Testing Method which will be explained further in chapter 3.

## CHAPTER 2

### LITERATURE REVIEW

#### 2.1 Introduction

This chapter will cover few important theories related to the project. It is divided into few parts which consist of introduction on control valve, control valve stiction, and the existing statistical-based methods for fault detection. An overview for some of the chosen statistical-based techniques to be tested is also included under this section.

#### 2.2 Control Valve

In most of the literatures, control valves are known as the final control elements and the moving part in a process control loop. Once there is a flow of the process fluid, the valve will control the allowable volume required which can be seen at Figure 2.1 below. According to Fisher Controls International (2005), typical assembly of a control valve comprises of the valve body, the internal trim parts, the actuator and also accessories which includes the position sensors [5]. An ideal valve will have a linear flow characteristic with no dead time and properly adjusted packing [6].

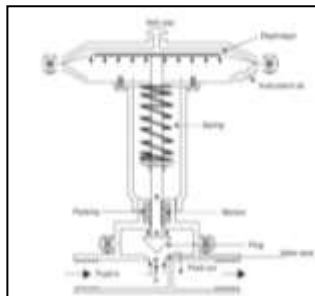


Figure 2.1 Cross-sectional diagram of control valve [7]

In general, there are four types of control valve where any possible of bad occurrences are possible to happen at any control system [8]. The categories of the control valve are listed in Table 2.1 below.

Table 2.1 Four general types of valves [8]

No	Types	Description
1	Electronic or electrical valves	The movement of the ball that controls flow is controlled electronically through circuits
2	Non-return valves	The valves only allow the flow in one direction
3	Electromechanical valves	These valves have electromagnets controlling to open or close the valve
4	Mechanical Valves	These valves use mechanical energy in the process of closing and opening of the valves such as pulleys and levers

There are four commonly used valves which are the ball valves, butterfly valves, globe valves and plug valves. Each type of valves has different functions depending on the application of the conditions. Despite of their types, any valve can suffer mechanical problems such as noise, cavitation, hysteresis and also valve stiction.

### 2.3 Control Valve Stiction

Same like other mechanical devices, control valve requires regular maintenance after certain period of its operation. There are many problems occurred which will then might affect the performance of the control loop. An initiative done by Saudi Aramco for process control improvement involving approximately 15,000 Proportional-Integral-Derivative (PID) loops, 50 Multivariable Predictive Control (MPC) and 500 smart positioners had proved that the common reasons for poor control loop performances are due to the valve stiction, hysteresis and backlash amongst others [9]. However, according to Bakri, et al (2014), valve stiction is the main culprit for the bad performance in the control loops [9]. Figure 2.2 illustrates the block diagram of closed loop control with sticky valve.

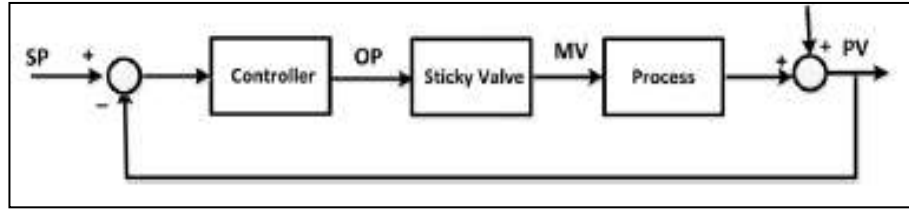


Figure 2.2 Typical closed loop control with sticky valve

Stiction has been described with different terms by many people and organizations. According to Choudhury, Shah, & Thornhill (2008), the sticky valve will illustrate four components which are deadband, stickband, slip-jump and moving phase which are shown in the Figure 2.3 [6].

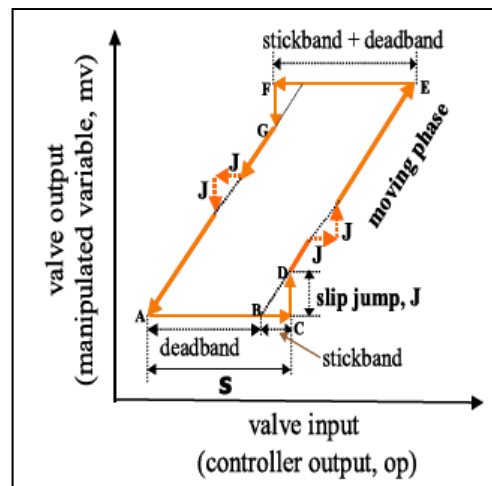


Figure 2.3 Typical behavior of a sticky valve [6]

Joe Qin (1998) mentioned that the stiction detection method was first initiated by Harris in 1989 which is the Control Performance Monitoring (CPM) by using the data of closed-loop to evaluate and diagnose controller performance using a minimum variance control [10]. From that time, many people started to develop or redesign the stiction detection methods to fit with different situations of the final control element. Table 2.2 further describes control valve stiction detection methods that have been implemented from 1999 up until 2015[4, 11-18].

Table 2.2 Stiction detection methods with descriptions

No	Author (Year)	Method	Description
1	H. Zabiri M. Ramasamy (2009)	Nonlinear Principal Component Analysis (NLPCA)	<ul style="list-style-type: none"> <li>• Calculates an index based on NLPCA using the distinctive shapes of the signals</li> <li>• Utilizes the controller output (pv-op)</li> <li>• Regression coefficient (<math>R^2</math>) is more than 0.8 whereby it includes a curvature index, INC</li> <li>• It will help to distinguish between the shapes of the signals caused by stiction and other sources</li> <li>• It is also known as neural-network based generalization of PCA</li> </ul>
2	Choudhury et al (2008)	Bicoherenc (automatic method to detect nonlinearities)	<ul style="list-style-type: none"> <li>• Involve the calculation of higher order statistics of the closed-loop data</li> <li>• It has two indices which are non Gaussianity index (NGI) and nonlinearity index (NLI)</li> </ul>
3	Singal & Salsbury (2005)	Calculation of Ratio of Areas, R	<ul style="list-style-type: none"> <li>• Based on calculation of areas before and after the peak of an oscillating signal</li> <li>• As stiction is detected, the R value is more than 1.</li> </ul>
4	Jelali et al (2008)	Optimization- based technique	<ul style="list-style-type: none"> <li>• Utilizes two main components which are Pattern Search (PS) and Genetic Algorithms</li> <li>• More towards stiction estimation</li> </ul>
5	M.A Daneshwar, Norlaili Mohd Noh (2015)	Well-developed fuzzy clustering approach	<p>A new index is proposed and tested</p> <p>The steps are:</p> <ol style="list-style-type: none"> <li>1) Find cluster centres using well-developed fuzzy clustering</li> <li>2) Fit a line to the obtained cluster centres</li> <li>3) Calculate the error of fitting</li> <li>4) Decision making using the obtained error with predefined threshold</li> </ol>
6	A. Horch (1999)	Shape-based Method	<ul style="list-style-type: none"> <li>• Based on the cross-correlation between the controller input and the process output signals.</li> </ul>



			<ul style="list-style-type: none"> <li>It is limited due to the flow loop dynamics which can distort the shape of the stiction pattern</li> </ul>
7	Babji Srinivasan, T. Spinner, Raghunathan Rengaswamy (2012)	Frequency Domain Analysis	<ul style="list-style-type: none"> <li>Use Hammerstein-based model</li> <li>This identifies a nonlinear stiction element along with a linear dynamical model between the process output and controller output</li> <li>The differences are calculated using Root Mean Squared Error (RMSE)</li> </ul>
8	S. Kalaivani, T. Aravind, D. Yuvaraj (2014)	A Single Curve Piecewise Fitting Method	<ul style="list-style-type: none"> <li>Based on qualitative analysis of the control signals</li> <li>To fit two different functions, triangular wave and sinusoidal wave, to the controller output data</li> </ul>
9	Bartys, M., Quevedo, J. Patton, RJ (2003)	Development and Application of Methods for Actuator Diagnosis in Industrial Control Systems (DAMADICS)	<ul style="list-style-type: none"> <li>Real time training of actuator system</li> <li>The testing was performed by inducing abrupt (sudden) and incipient (gradually developing) faults to the actuators and recording data</li> </ul>

From Table 2.2 shown above, a simple graph is plotted and only two out of nine methods developed used the statistical-based methods to detect the control valve stiction. Figure 2.4 illustrates the comparison of methods described above based on the recentness of the literatures.

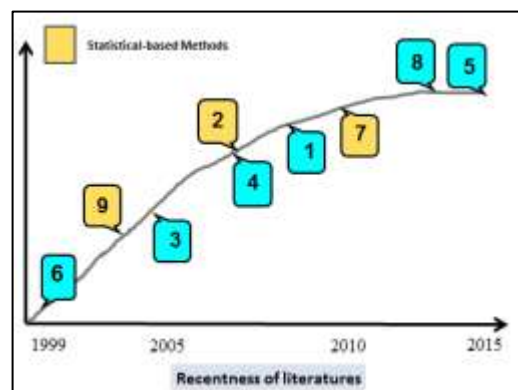


Figure 2.4 Comparison of stiction detection methods

## 2.4 Statistical-based Methods for Fault Detection

Fault Detection can be considered as the unpermitted deviation which can lead to the failure or malfunction in an engineered system [19]. There are several methods to detect and diagnose these faults by using either quantitative or qualitative models and also by referring to the large history data from the samplings. Srinivas et al (2005) stated that one of the history-based methods is the statistical-based method which is illustrated in Table 2.5 below:

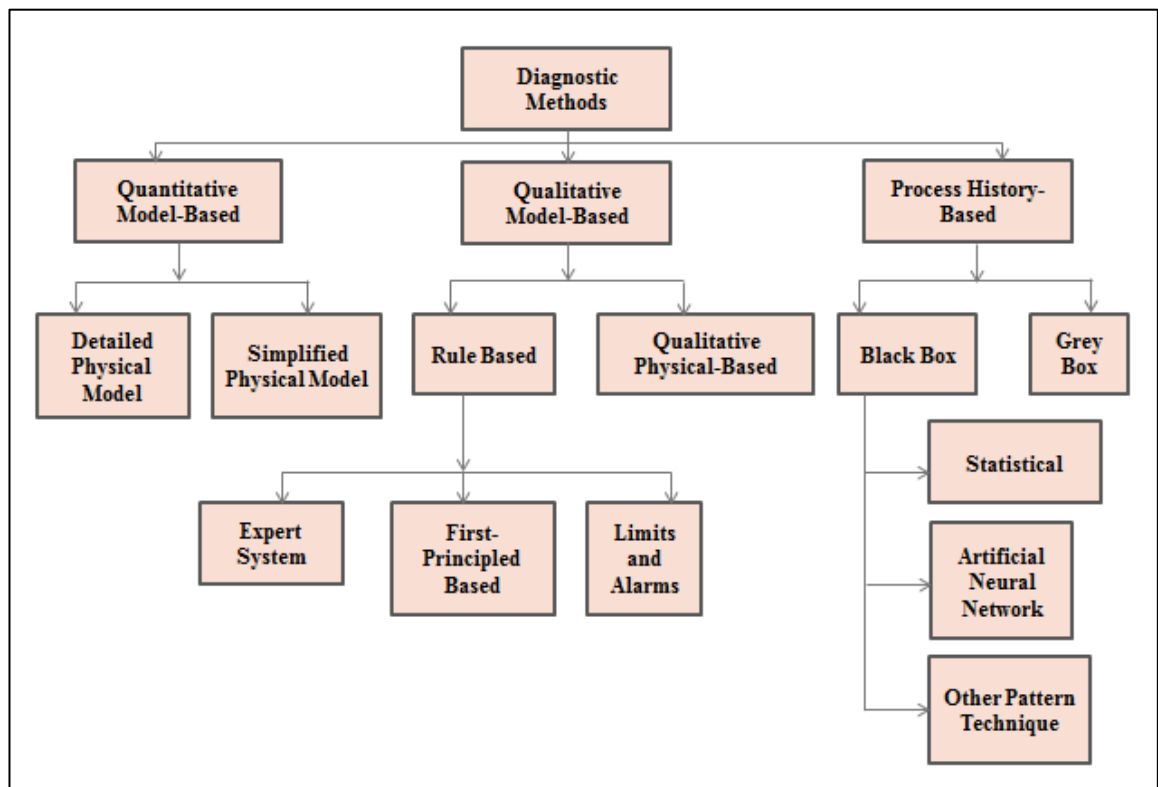


Figure 2.5 Classifications Scheme for Fault Detection [20]

Statistical method can be described as the observations made or characteristics that are inferred based on the sample from the entire populations or data. Since it is based on historical data from processes, the authors described this method and model as easy to be developed for any kind of faults. Meanwhile, L. B. Ang et al (2014) stated that additional requirement for this method is the knowledge on computational methods to run and evaluate the testing [21]. Some of the methods are described in Table 2.3[21-26].

Table 2.3 Fault detection methods using statistical-based method

No	Author (Year)	Method	Description
1	Hector J. Galicia Q. Peter He Jin Wang (2011)	Statistics Pattern Analysis (SPA)	<ul style="list-style-type: none"> <li>A new multivariate statistical monitoring framework. This approach is proposed when principle component analysis (PCA) method had some limitations. The difference is where PCA monitors process variables while SPA monitors the statistics of the process variables. There are two major steps which are: <ol style="list-style-type: none"> <li>Statistics Pattern (SP) generation</li> <li>Dissimilarity quantification</li> </ol> </li> <li>In the fault diagnosis, we apply PCA to quantify the dissimilarity</li> </ul>
2	Fouzi Harrou, Mohamed N. Nounou, Hazem N. Nounou, Muddu Madakyaru (2012)	PCA-based Generalized Likelihood Ratio (GLR) Fault Detection Algorithm	<ul style="list-style-type: none"> <li>A statistical hypothesis testing method. This algorithm provides optimal properties by maximizing the detection probability of faults for a given false alarm rate</li> <li>There are two types which are simple hypothesis (depends on existence of faults) and composite hypothesis (assume a parameter as benchmark)</li> <li>GLR approaches replace the unknown parameter by its maximum likelihood estimate</li> <li>Advantage of this method is it can handle the presence of noise and uncertainties</li> </ul>
3	Carl Svard, Mattias Nyberg, Erik Frisk, Mattias Krysander (2013)	Statistical Residual Evaluation with an automotive application	<ul style="list-style-type: none"> <li>This is where residuals are evaluated with the aim to detect changes in their behaviors caused by faults</li> <li>The framework used is the hypothesis testing</li> <li>Input is the set of residual samples and output is the decision to reject the hypothesis or not</li> </ul>

4	Liu Bo-ang, Ye Hao (2014)	Statistical $\chi^2$ testing for linear discrete time-delay systems	<ul style="list-style-type: none"> <li>The constructed function used the measurements observations. It used projection and innovation analysis</li> <li>i. First step – generate JN (evaluation function) online</li> <li>ii. Second step - <math>\chi^2</math> statistical testing</li> <li>The strategies are single observation, observation sequence, window average of observations</li> <li>The important remark is that the process and measurement noises being with jointly normal distribution</li> </ul>
5	Yang Zhao, Shengwei Want, Fu Xiao (2012)	Exponential- Weighted Moving Average (EWMA) control charts with Support Vector Regression (SVR)	<ul style="list-style-type: none"> <li>Combining EWMA control charts with SVR</li> <li>i. EWMA – to detect faults in statistical way to improve the ratios of correctly detected points</li> <li>ii. SVR – to improve accuracy of the reference models</li> <li>Case study showed that this method improved the FDD performances especially at low severity levels</li> </ul>
6	Ashwani Kumar & A. P. Singh (2013)	Polynomial Curve Fitting	<ul style="list-style-type: none"> <li>It involves with circuit simulation</li> <li>This method used artificial neural networks as the case study for the method</li> <li>The results of the fitting are calculation of residuals, least squares, residuals and other</li> </ul>

From the listed statistical-based methods that have been tested on fault detection, two methods are picked to test the capability on detecting control valve stiction based on the generated op-pv data which are generalized likelihood ratio (GLR) test and polynomial curve fitting method.

## **CHAPTER 3**

### **METHODOLOGY**

#### **3.1 Introduction**

This chapter includes the detailed methods in achieving the objectives of this study. It starts with general project flow chart which covers the data of process output (denoted as pv) and controller output (denoted as op) from the case study or industrial data developed based on Nonlinear Principle Component Analysis (NLPCA) method as well as the statistical-based techniques steps. Besides that, project gannt chart and key milestones for both semesters are included together with the tools used for this project.

#### **3.2 General Project Work**

The process flow for this research project is divided into five stages which are shown in Figure 3.1. Based on the previous chapter, the literature review also covers few options of statistical-based methods for fault detection. From Table 2.3 which covers the existing fault detection techniques using statistical-based methods, two techniques are chosen to be tested with the generated data with some modifications on the methods. Furthermore, both of the methods are tested with real industrial data from existing process plant for better justifications.

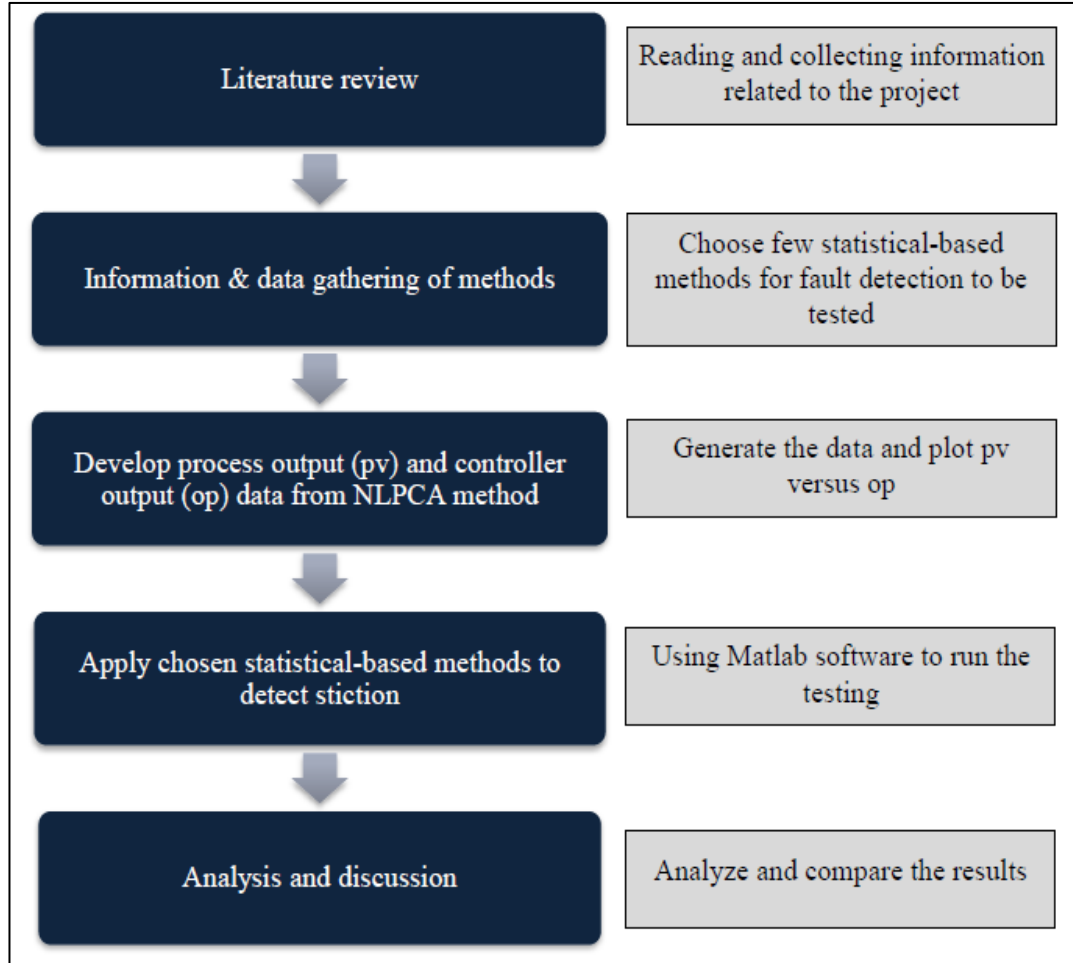


Figure 3.1 Process flow chart

### 3.2.1 Development of pv and op data

#### 3.2.1.1 Simulation case study

In order to test on the effectiveness of the statistical-based method of fault detection in detecting control valve stiction, a case study had been chosen with some operating parameters involving stiction and non-stiction conditions. A simple single-input and single-output (SISO) feedback control system is used to generate the simulated data. The transfer function of the process is:

$$G(z^{-1}) = \frac{z^{-3} x (1.45z - 1)}{z - 0.8} \quad (1)$$

The process was assumed to be linear and controlled using the Proportional Integral (PI) controller. Each four cases are described based on the process output and the controller output. The simulated results were produced based on nonlinear principal component analysis (NLPCA) done by Zabiri and Ramasamy [4].

#### Base case : Well-tuned controller

The PI controller is designed with the controller gain of  $K_c = 0.15$  and integral time  $\tau_I = 0.15$ . It is considered as well-tuned controller because of the high stability of the system with no presence of stiction and external oscillatory disturbances. The block diagram for the base case is as follows in Figure 3.2:

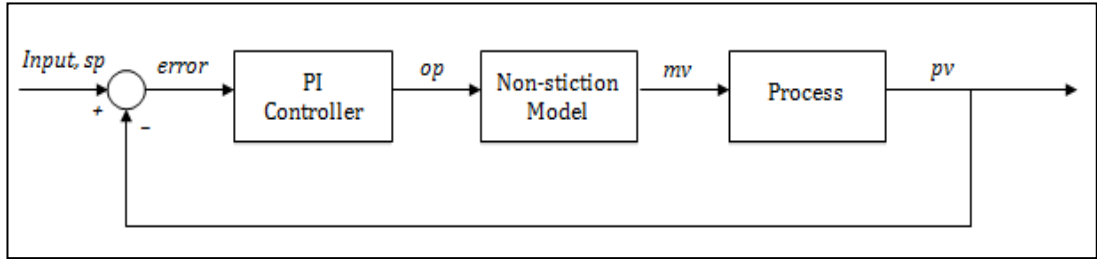


Figure 3.2 Block diagram for well-tuned controller

#### Case 1 : Tightly-tuned controller

For case 1 controller, the PI controller operating parameters are  $K_c = 0.15$  and  $I = 0.27s^{-1}$ . There are no stiction or external oscillatory disturbances action towards the system. However, the integral time increases which make the integral action disturb the output from the controller. The block diagram is shown in Figure 3.3.

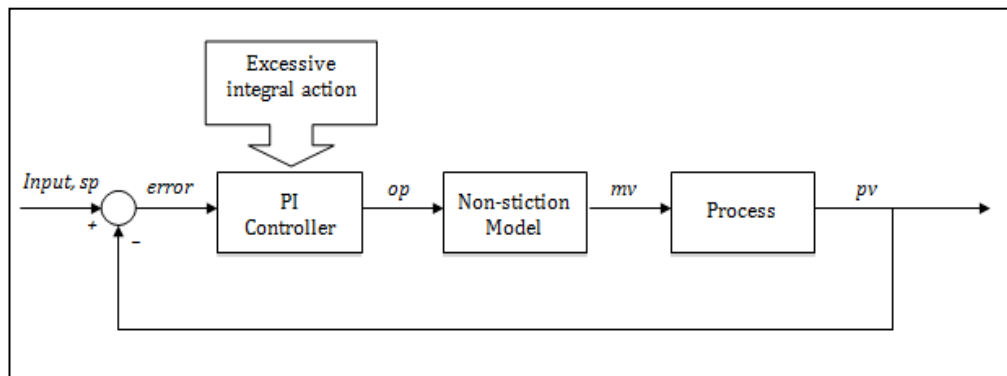


Figure 3.3 Block diagram for excessive integral action controller

## Case 2 : Existence of external oscillatory disturbances

Similar to base case condition, the parameters are designed with the same values for controller gain and integral time. However, a sinusoidal disturbance with amplitude of 2 and a frequency of 0.01 is added to the process which resulted with a disturbance action towards the performance of the control loop. The block diagram in Figure 3.4 further describes the condition of case 2.

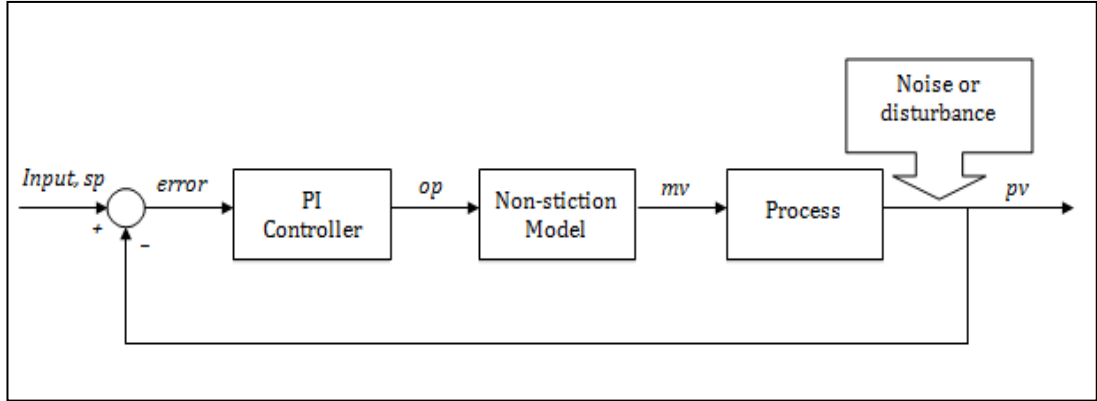


Figure 3.4 Block diagram for external oscillation controller

## Case 3 : Presence of stiction

In this case, the simulation is done with the presence of stiction in the control loop. The stiction method proposed will then prove the existence of stiction in the process loop. The stiction model is shown in Figure 3.5 after the PI controller.

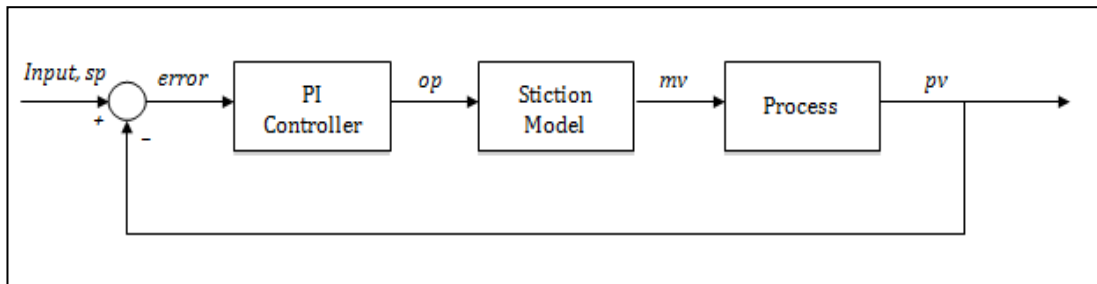


Figure 3.5 Block diagram for control system with presence of stiction

### 3.2.1.2. Industrial case study

After being tested with the simulation case study, the evaluation is continued by applying the methods to different kind of industrial loops. The performance of the chosen methods are then been applied with some real data from flow control loops



which are based on fuzzy clustering data [14]. Table 3.1 had simplified the flow control loops chosen.

Table 3.1 Industrial data from flow control loops

Flow Loops	Number of data	Conditions
Loop 1	1441 samples	Evident absence of stiction
Loop 2	1998 samples	Presence on stiction
Loop 3	200 samples	Presence of stiction
Loop 4	719 samples	Presence of external disturbances
Loop 5	721 samples	Presence of external disturbances

### 3.2.2 Statistical-based methods steps

Based on the statistical methods for fault detection which had been described earlier in chapter 2, the chosen methods to be tested are generalized likelihood ratio (GLR) test as well as error testing method.

#### 3.2.2.1 Generalized likelihood ratio (GLR) test

To implement the GLR test with the data generated from the case study, a statistical hypothesis testing was adopted. Since the main objective of this fault detection system is to detect any existence of stiction, the test had come out with two decisions which are null hypothesis (indicates stiction free condition) and alternative hypothesis (indicates stiction exists). The equation for the GLR,  $\mathcal{L}(R)$  test was computed using equation (2) below;

$$\mathcal{L}(R) = \frac{1}{\sigma\sqrt{2\pi}} \exp^{\frac{-1}{2\sigma^2} \|r-\mu\|_2^2} \quad (2)$$

From the above equation,  $\sigma$  is defined as the standard deviation,  $\sigma^2$  is the variance,  $r$  is the residual and  $\mu$  is the mean from the data generated. With a significance level of 5% and one degree of freedom, the decision threshold value is chosen to be 3.841 for both case studies. Hence the method can detect and report the existence of stiction if  $\mathcal{L}(R)$  is bigger than the decision threshold value. For non-

stiction system, the null hypothesis is accepted which indicates stiction free condition.

To summarize the steps involves, Figure 3.6 is developed which explained the actions to run this method for the case study.

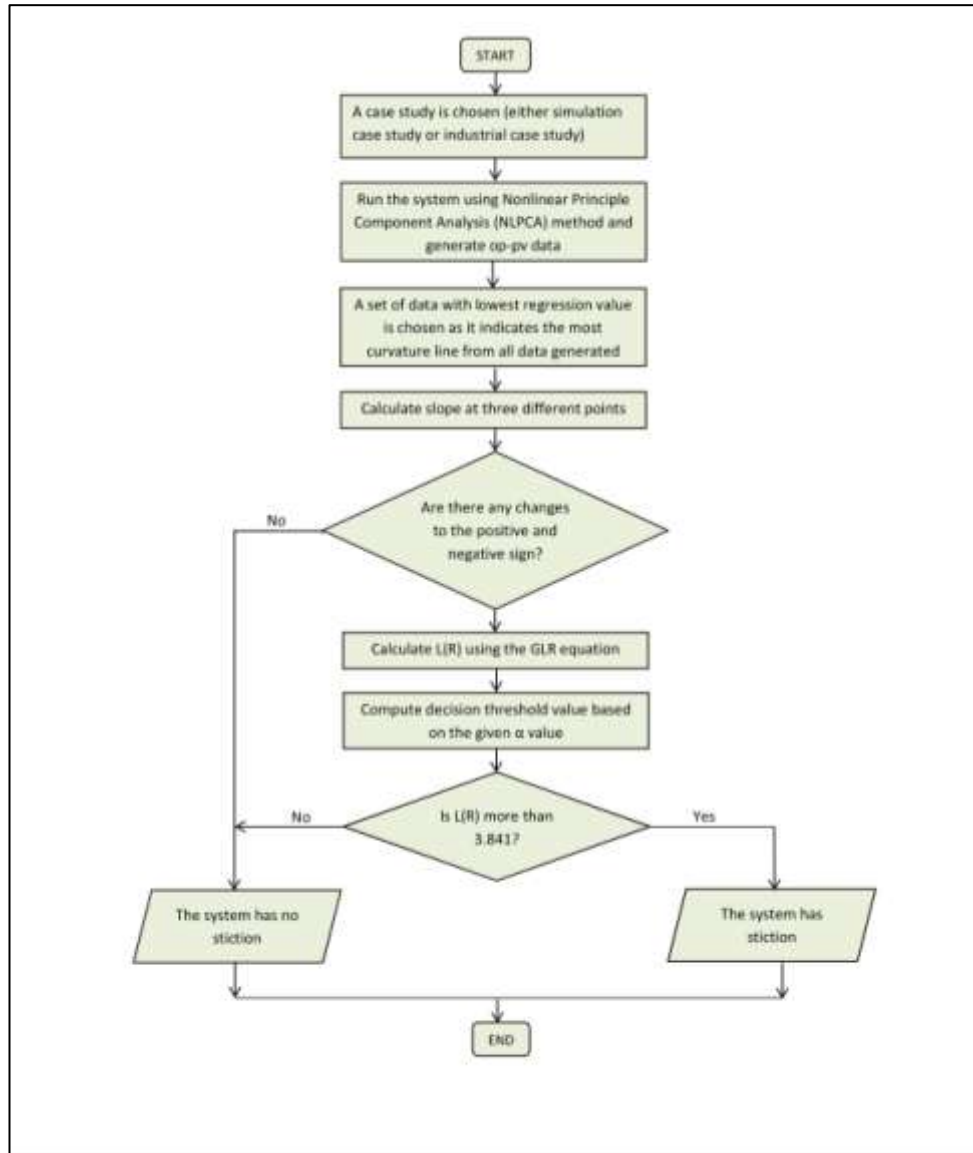


Figure 3.6 Flowchart of GLR testing method

### 3.2.2.2 Error Testing Method

For error testing method, the fitting known as curve fitting is applied whereby the generated data from the case studies are fitted to obtain few parameters which are

sum of the squared error (SSE), regression ( $R^2$ ) as well as root mean squared error (RMSE). All of these can be obtained by applying the Curve Fitting Toolbox ('cftool' command) in Matlab software for fitting curves and generates the data [27]. Additional steps are made which involve with the calculation of slope at three different points. For each slope values, any changes in the positive and negative values will indicates the curve shapes and can be said as stiction system. Figure 3.6 shows the Curve Fitting Toolbox used in this project.

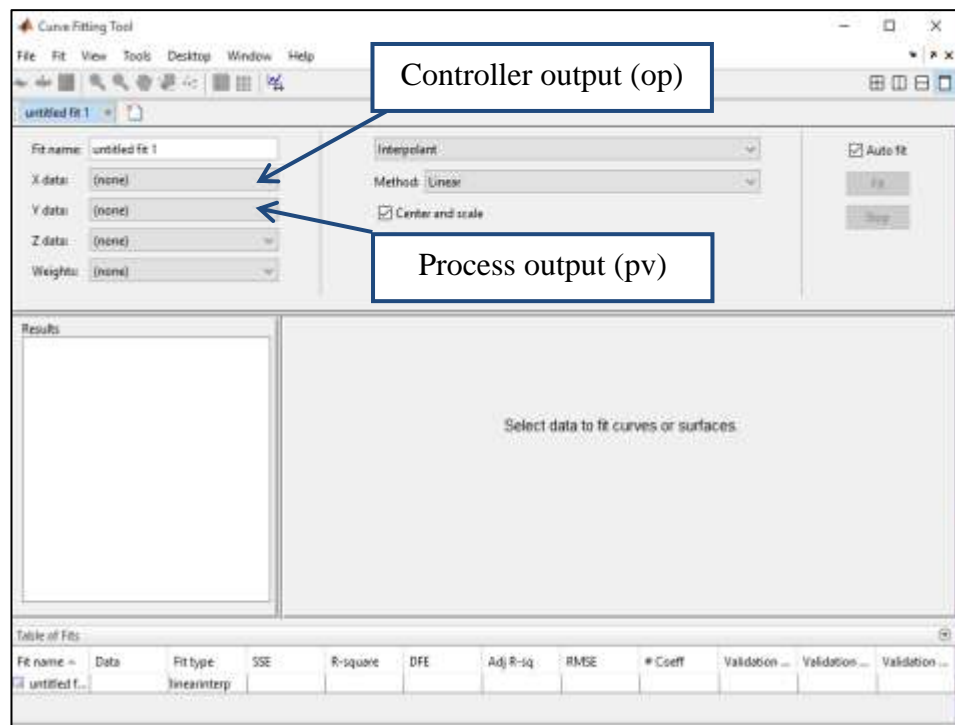


Figure 3.7 Curve Fitting Toolbox in Matlab

Figure 3.8 further illustrates the steps to be applied in error testing method for the case study.

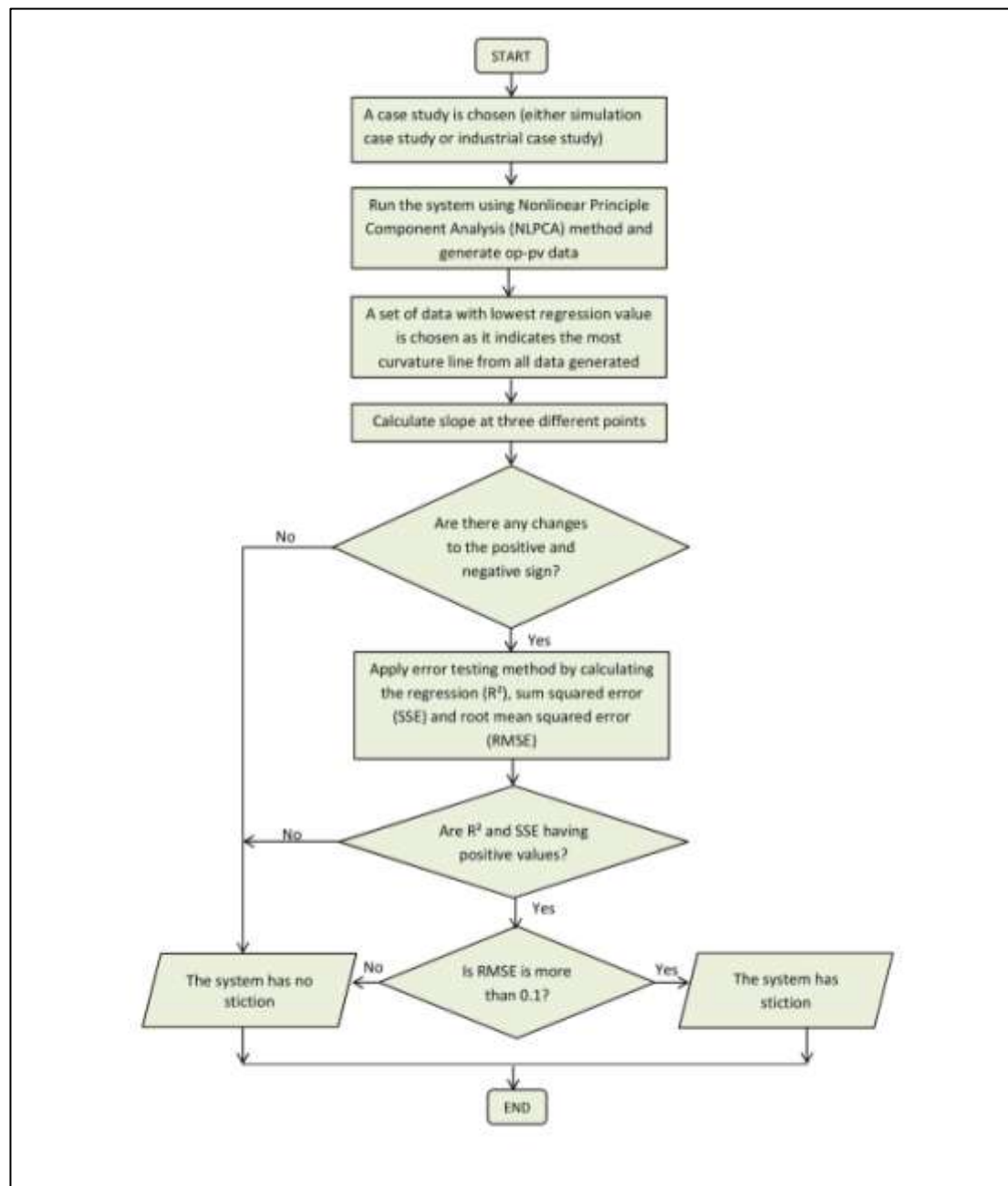


Figure 3.8 Flowchart of error testing method

### 3.3 Project Gantt Chart and Key Milestone

Throughout the two semester of completing the project, there are two project gantt charts which act as the main guideline for every work as shown below:

Table 3.2 Project gantt chart and key milestone

Final Year Project (Semester 1)															
No	Detail Work	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	Selection of Project Topic														
2	Preliminary Research work and Literature Review														
3	Submission of Extended Proposal (first draft)														
4	Submission of Extended Proposal (final draft)														
5	Preparation for Proposal Defense														
6	Proposal Defense														
7	Project Work Continues														
8	Submission of Interim Draft Report														
9	Submission of Interim Report														
Final Year Project (Semester 2)															
No	Detail Work	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	Selection of Project Topic														
2	Submission of Progress Report														
3	Project Work Continues														
4	Pre-SEDEX														
5	Submission of Draft Final Report														
6	Submission of Dissertation (soft bound)														
7	Submission of Technical Paper														
8	Viva														
9	Submission of Project Dissertation (Hard bound)														

Gantt chart

Key Milestone



Gantt chart



Key Milestone

### **3.4 Tools**

The tools used are software tools which are mainly used for simulation, testing and documentation purposes.

- a. MATLAB – this tool is used for simulation purposes
- b. Microsoft Office Words – this tool is mainly used for writing such as the proposal, interim report and full report
- c. Microsoft Office Excel – this tool is used to develop any tables from the generated data for analysis purposes

## **CHAPTER 4**

### **RESULTS AND DISCUSSION**

#### **4.1 Introduction**

This chapter describes on the results obtained based on the application of the two selected statistical-based methods with both case studies of simulation and real industrial data respectively. The first part presents the outcomes from the GLR stiction detection method followed by the error testing method. The results are well analysed and discussed in the following sections together with the constraints if any.

#### **4.2 Generalized Likelihood Ratio (GLR) Test**

For this first chosen statistical-based method, a closed-loop feedback system with a process model which uses equation (1) as mentioned earlier is constructed in Matlab software. It will be used to generate the simulated data. The Simulink blocks for each case are attached in *Appendix A*. The process was assumed to be linear and controlled using the PI controller. Each four cases under this simulation case study which have been described earlier in chapter 3 will produce different set of process output (pv) and controller output (op) data based on the nonlinear principle component analysis (NLPCA) method[4].

For base case which is considered as the well-tuned controller in Figure 4.1 below, it shows the graph plotted after 10 runs from NLPCA technique.

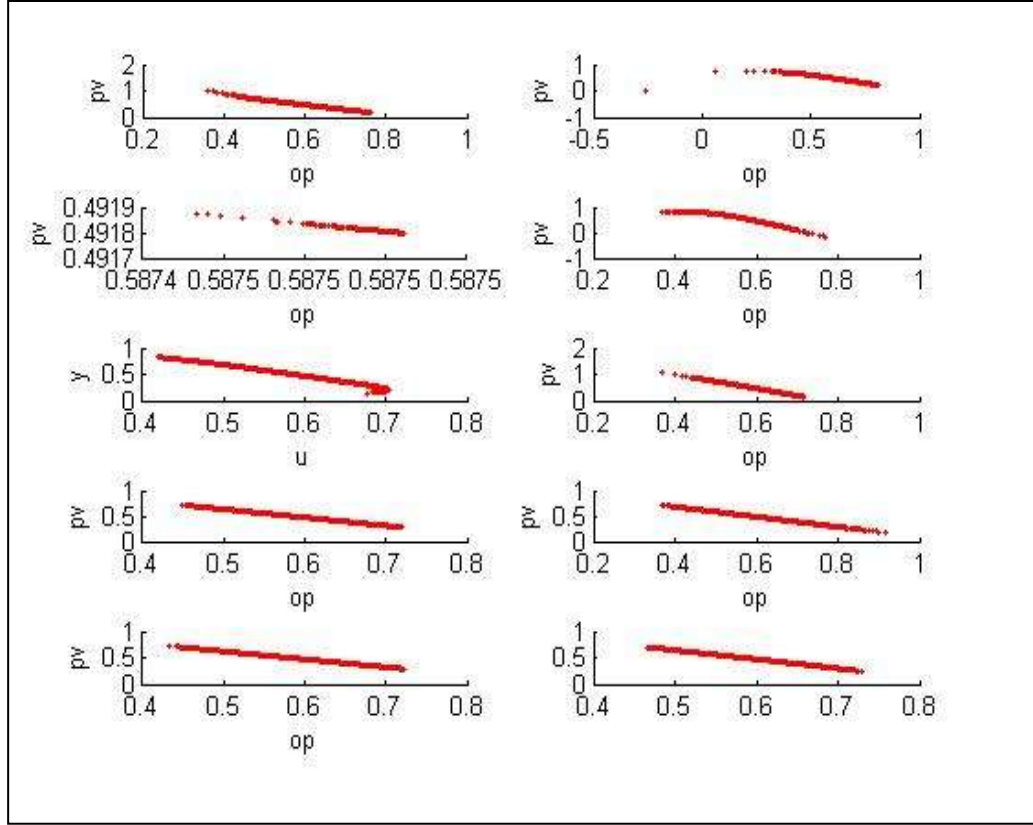


Figure 4.1 The pv-op plotted graph for base case

As expected, the op and pv system generated for base case is in straight line without any curvature shape. The system is more stable without oscillation. In order to get the best fit curve from the plotted graph, calculations of regression are made. With the help from Matlab software, the command of *polyfit* and *polyval* have helped the author in selecting the lowest regression value from all the data generated which will be chosen for further quantification of stiction with GLR test. Case 1, 2 and 3 also undergo NLPCA method and the shape generated for the 10 times run are shown in Figure 4.2, Figure 4.3 and Figure 4.4. For the lowest regression values based on op and pv data, the chosen graph are also included in Table 4.1.



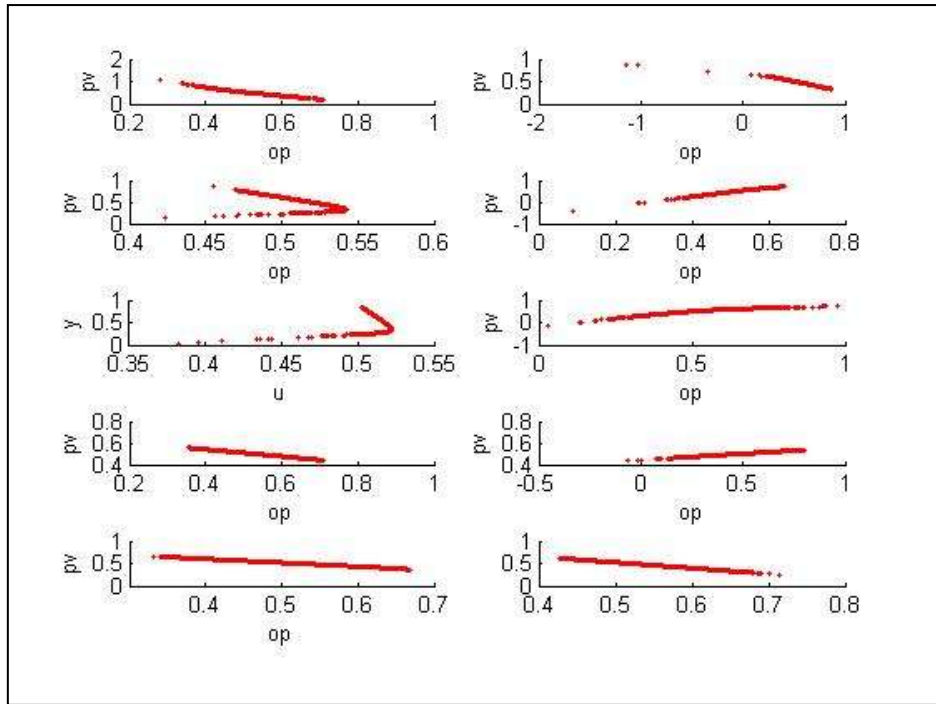


Figure 4.2 The  $pv$ - $op$  plotted graph for case 1

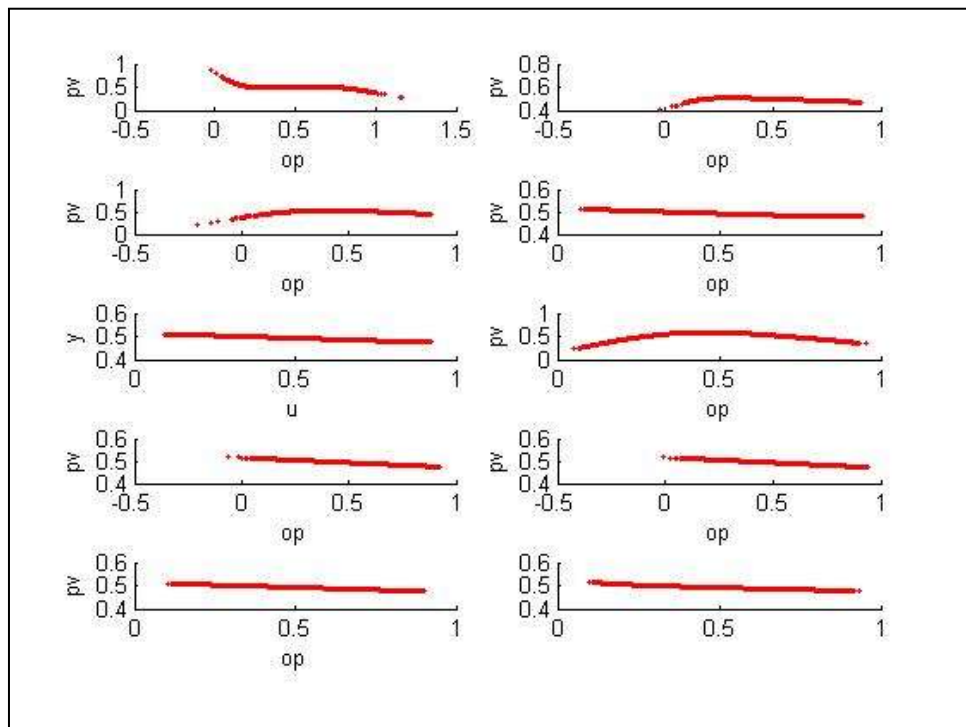


Figure 4.3 The  $pv$ - $op$  plotted graph for case 2

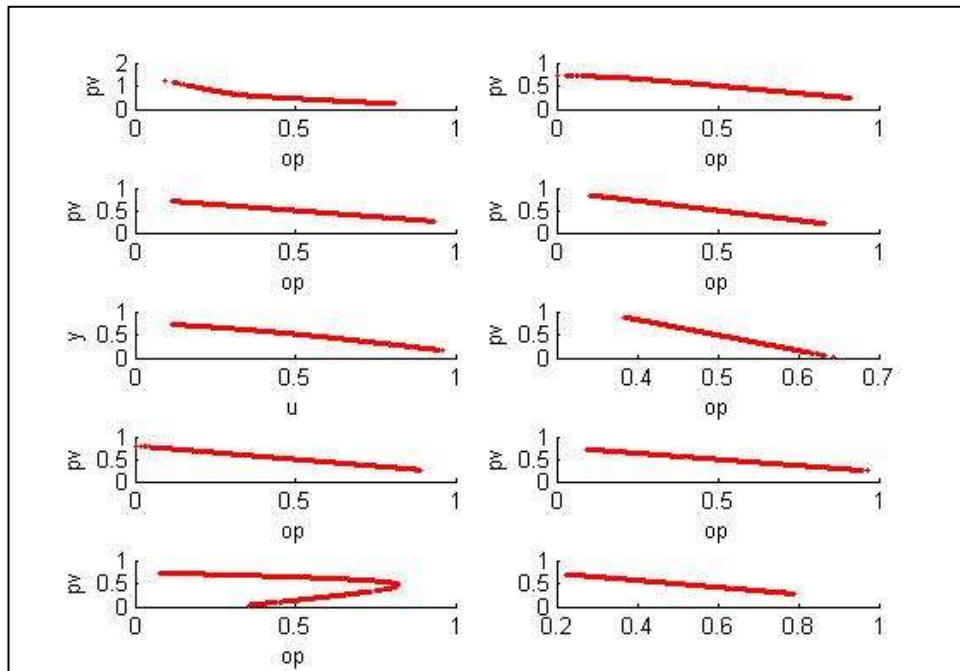


Figure 4.4 The pv-op plotted graph for case 3

Table 4.1 The chosen pv and op data for lowest regression values

Base case Well-tuned controller	Case 1 Tightly-tuned controller
Case 2 Presence of oscillation	Case 3 Presence of stiction

The ability of GLR test in detecting stiction for each case study is investigated and further discussed. After choosing the set of pv and op data based on the lowest regression values calculated in Matlab software, the GLR test statistic value is tested using equation (2) mentioned in section 3.1.2.1 earlier. The testing was done using the method of residual calculation. The probability of significance level used is 5% and the chi-square value for 5% with 1 degree of freedom is 3.841. The GLR test statistic value is compared with the chi-square value to detect the existence of valve stiction. For values that are lower than 3.841, it indicates no stiction and higher than 3.841 indicates the existence of stiction in the loop. Hence, Table 4.2 summarized the results of the statistical-based method.

Table 4.2 Summary of results generated for all case studies

Case Study	GLR Test Statistic Value	Detection Result
(1) well-tuned controller	0.1152	The system has no stiction
(2) tightly-tuned controller	0.9289	The system has no stiction
(3) presence of external oscillatory disturbances	7.8129	Fault present. The system has stiction
(4) presence of stiction controller	11.9713	Fault present. The system has stiction

Based on the results of the testing, it can be seen that three out of four cases resulted in the correct output. However, case study 3 which involves with the presence of oscillatory disturbances produced higher value than the chi-square value, 3.841. It is then accidentally shows the existence of stiction to the system. This actually indicates the error of calculations and input for the case study.

Hence, based on the above results, it is proven that the test is only applicable for base case, case 1 and case 3 conditions. For presence of external disturbances controller, it has a high possibility to be detected as stiction system due to the curvature and high value of GLR test statistic value.

### 4.3 Error Testing Method Test

For error testing method, similar procedures are applied for the generation and selection of pv and op data with the GLR testing method. Before applying the calculation, the cases are first been filtered out with the slope calculation. If there are some changes for the sign, it will indicate curvature shape which has high possibility with the presence of stiction in the system. Table 4.3 summarizes the results of the slope calculation.

Table 4.3 Slope calculations for simulation case study

Base Case			Straight line
Slope 1	Slope 2	Slope 3	
-1.2494	-1.2783	-1.2561	
Case 1			Straight line
Slope 1	Slope 2	Slope 3	
-22.4225	-26.2544	-23.0422	
Case 2			Curve shape
Slope 1	Slope 2	Slope 3	
-0.2092	0.9163	-0.6246	
Case 3			Curve shape
Slope 1	Slope 2	Slope 3	
0.8645	-0.2923	0.839	

It can be seen from Table 4.3 whereby for case 2 and case 3, it has a high possibility to be detected as stiction system. For base case and case 1, since there are no changes for the slope at three different points, it indicates straight line and there are absent of stiction in both of the system.

With the help of Curve Fitting Toolbox in Matlab, the op and pv data chosen is used to be plotted and calculated for the values of regression, sum squared error and root mean squared error. Figure 4.5 until 4.8 show the polynomial plot of all cases.

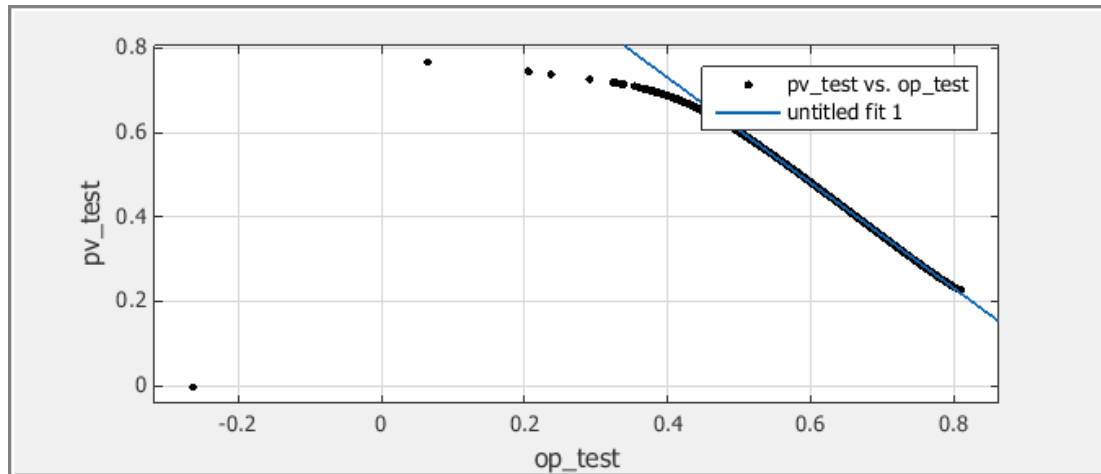


Figure 4.5 Polynomial plot for base case

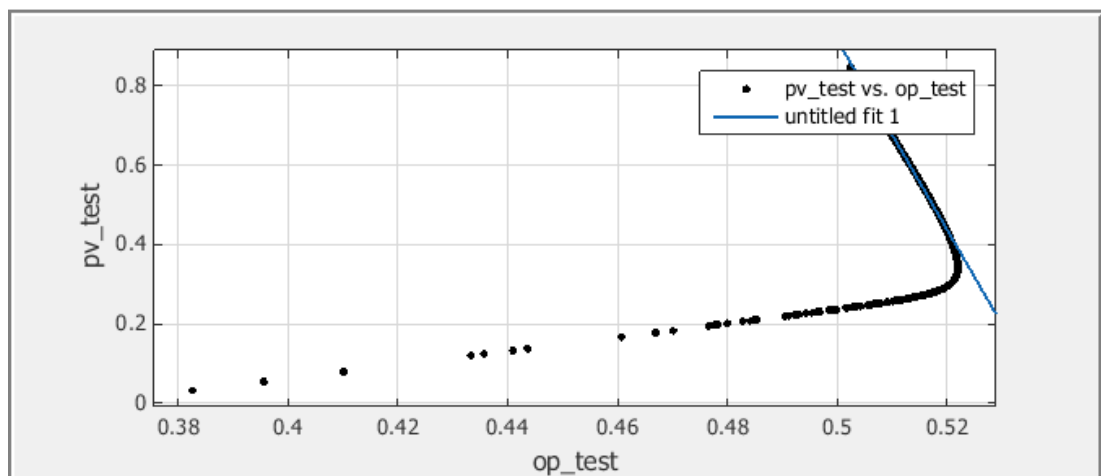


Figure 4.6 Polynomial plot for case 1

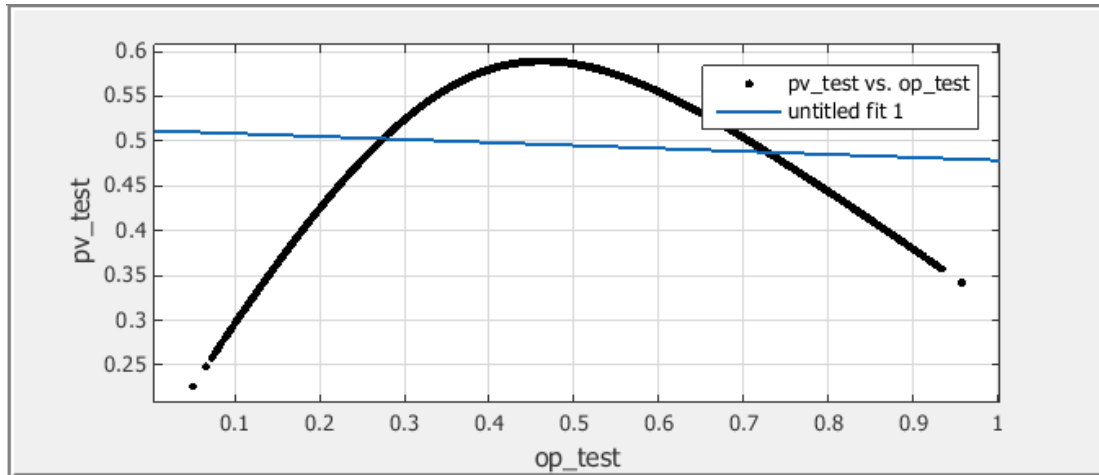


Figure 4.7 Polynomial plot for case 2

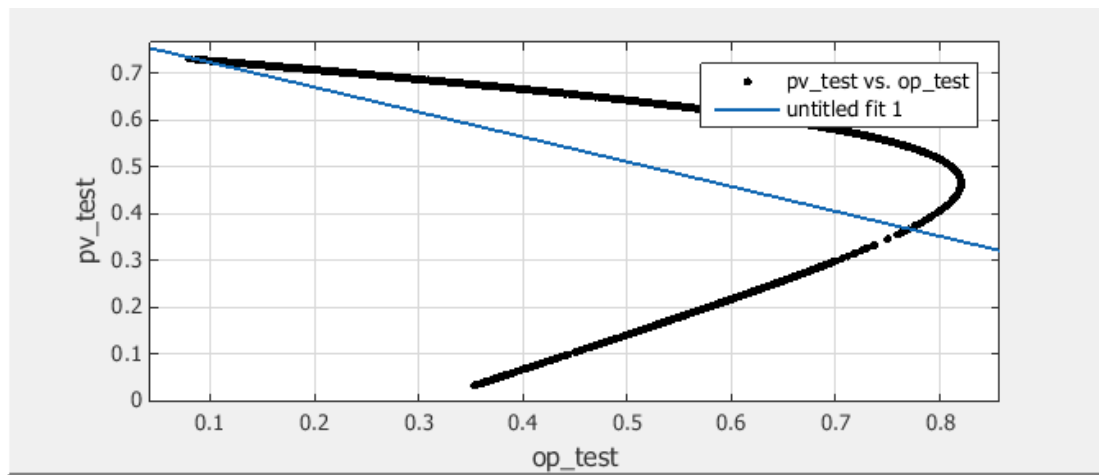


Figure 4.8 Polynomial plot for case 3

Based on the above figures, it can be seen that for well-tuned controller and tightly-tuned controller, the plotted polynomial curve contain some part of straight line which have been detected earlier in the slope calculation. Meanwhile for case 2 and case 3, both portrayed curvature shape but for stiction case, the curvature is higher and had different direction compared to case 2. The next part of this testing is the calculation of the values for regression ( $R^2$ ), sum squared error (SSE) and also root mean squared error (RMSE). Table 4.4 developed shows the summary of the calculations for all cases.

Table 4.4 Summary of error calculation for each case

Error Calculation	Base case	Case 1	Case 2	Case 3
$R^2$	0.9996	0.9976	-0.05131	0.3547
SSE	0.01501	0.1438	20.28	113.7
RMSE	0.001891	0.005852	0.0695	0.1645

Based on the data above, it can be seen that the regression value for base case and case 1 are approaching to 1. This shows that for these cases, the ideal and the actual have not much difference as for perfect case without stiction will have the regression value of 1. Similar thing for case 3 whereby the extreme curve portrayed in this situation had caused the regression value to be low which is 0.3547. However, for case 2 involving the external disturbances, there are some challenges as the regression value calculated is negative value. This shows that the regression calculation is sensitive to upward curvature. Hence, the calculation of regression to indicate stiction is not enough. Figure 4.9 shows the plotted graph of all the  $R^2$  values.

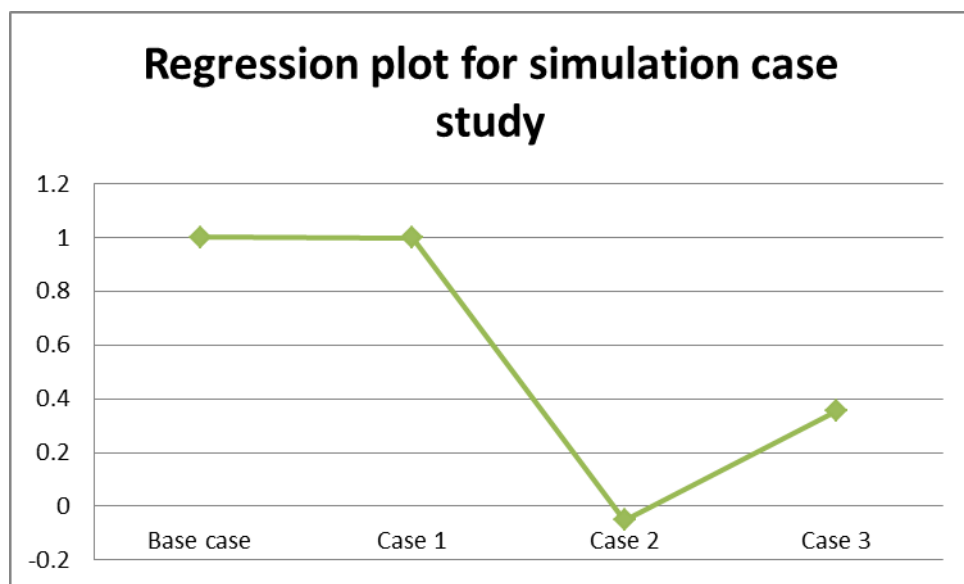


Figure 4.9 Comparison on the regression values

Meanwhile, the method is continued with the calculation of sum squared of error, SSE. In general, base case and case 1 had very small values of SSE which is less than 1. Meanwhile for case 2, the value of SSE is slightly higher and case 3 with stiction has the highest among others. Figure 4.10 shows the comparison for all SSE values for this case study. A simple hypothesis can be made that for stiction system, the values of SSE is the highest among other cases.

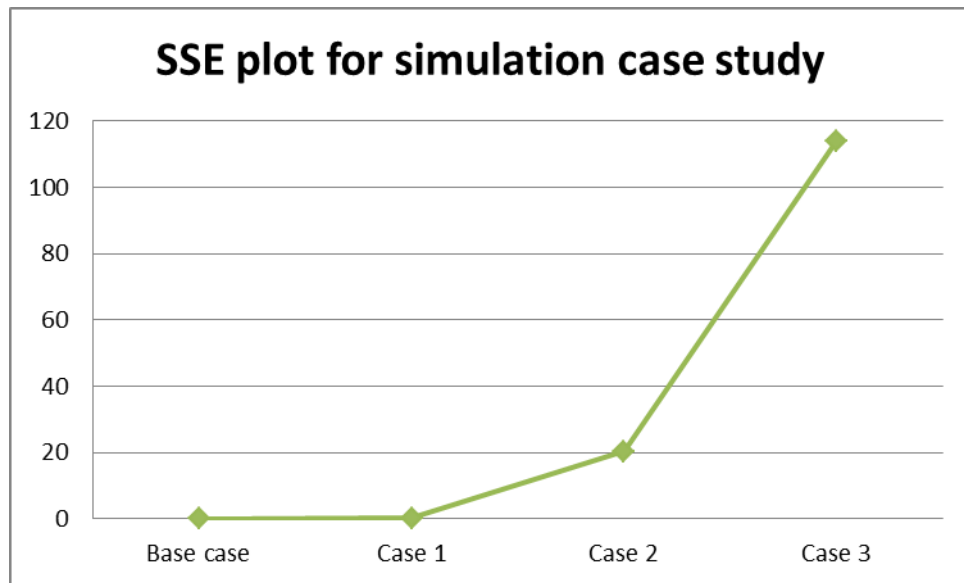


Figure 4.10 Comparison on the sum squared of error values

Another calculation of error is done which is the root mean squared of error, RMSE. According to the values generated in Matlab toolbox, similar to the previous error of SSE, the base case and case 1 have very small values of RMSE. Meanwhile case 2 has slightly higher which is 0.0695 which is smaller than 0.1. However, for stiction system, the RMSE is higher than 0.1 which is 0.1645. This can be considered for the comparison especially to differentiate between case 2 and case 3. Figure 4.11 represents the values of RMSE for all four case studies.



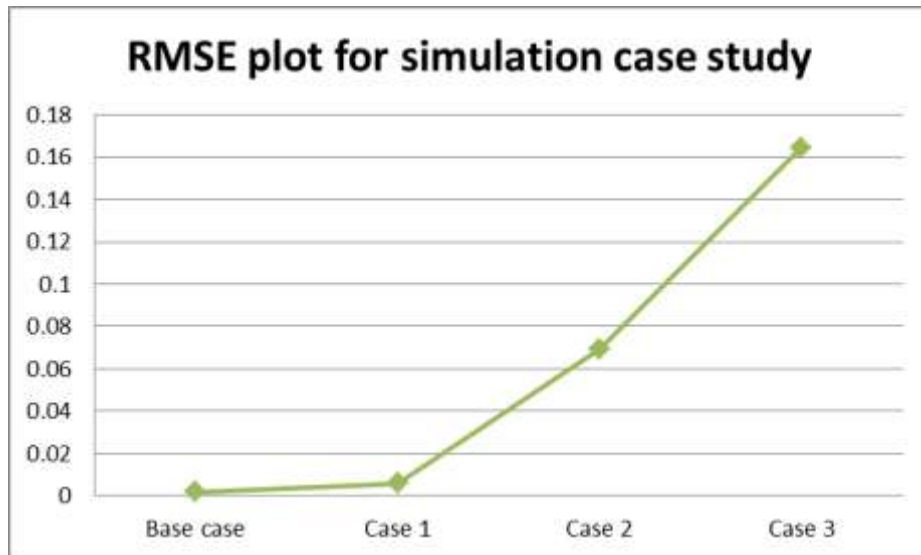


Figure 4.11 Comparison on the RMSE for all cases

Now the error testing method is further tested for the real industrial data. This is very important as the main goal for this statistical-based detection technique is to be able on detecting the stiction system for real plant operations. Similar steps are repeated whereby the industrial case studies undergo NLPCA method to generate the op and pv data. Then, the minimum regression values among the 10 run will be chosen as the pv and op data for the polynomial curve fitting.

The method starts with the slope calculation for each loop at three different points. Similar to Table 4.3, the slope calculations for the real industrial data and the indications for the sign changes are summarized in Table 4.5.

Table 4.5 Slope calculations for real industrial data

Loop 1			Straight line
Slope 1	Slope 2	Slope 3	
-1.6988	-1.6987	-1.6988	
Loop 2			Curve shape
Slope 1	Slope 2	Slope 3	
1.3240	-1.2879	-0.8842	

Loop 3			Straight line
Slope 1	Slope 2	Slope 3	
-1.3324	-1.3331	-1.3328	
Loop 4			Straight line
Slope 1	Slope 2	Slope 3	
-0.9470	-0.5177	-1.1225	
Loop 5			Straight line
Slope 1	Slope 2	Slope 3	
-0.0562	-0.2821	-0.2714	

It can be seen from the above table, the changes of sign happened in loop 2 which has a high possibility to be detected as stiction system. Meanwhile, for other loops, it indicates no changes to the sign which have the potential of straight line shapes generated from the NLPCA method. Hence, further steps are being made for better clarification.

From Figure 4.12 until Figure 4.16, these show the polynomial plots for each loop using the curve fitting toolbox in Matlab software.

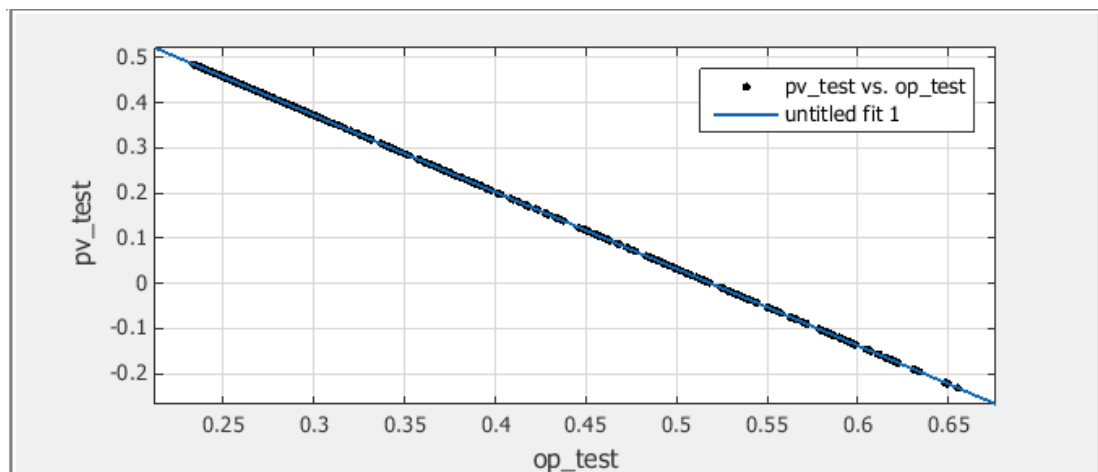


Figure 4.12 Polynomial plot for loop 1

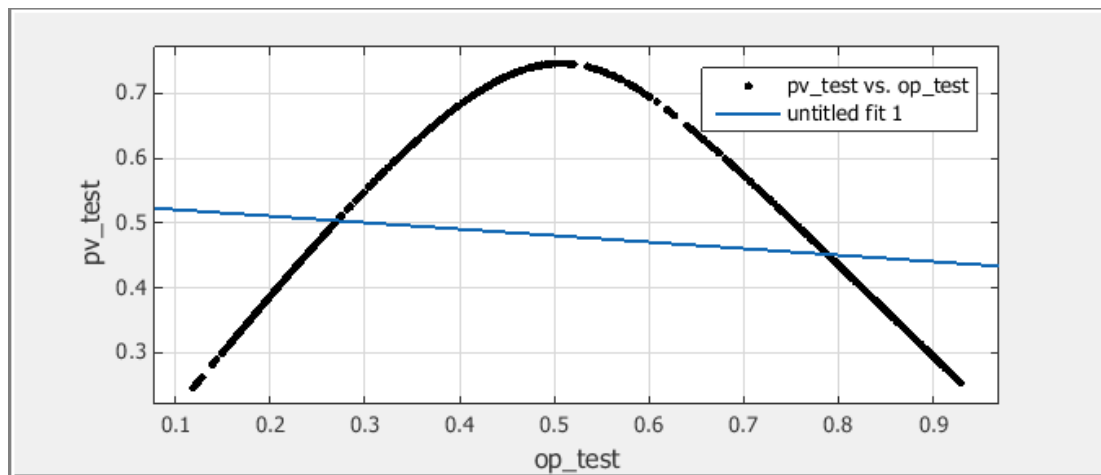


Figure 4.13 Polynomial plot for loop 2

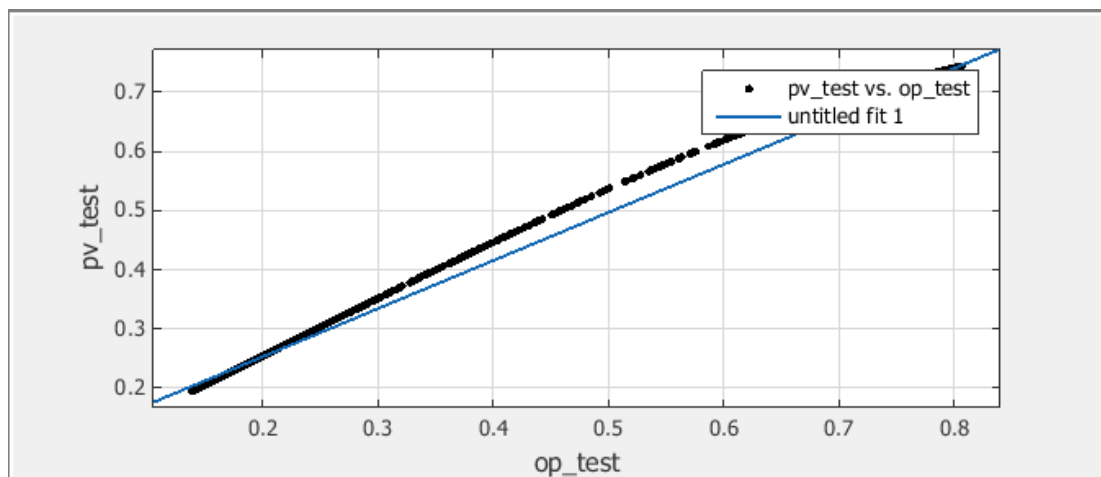


Figure 4.14 Polynomial plot for loop 3

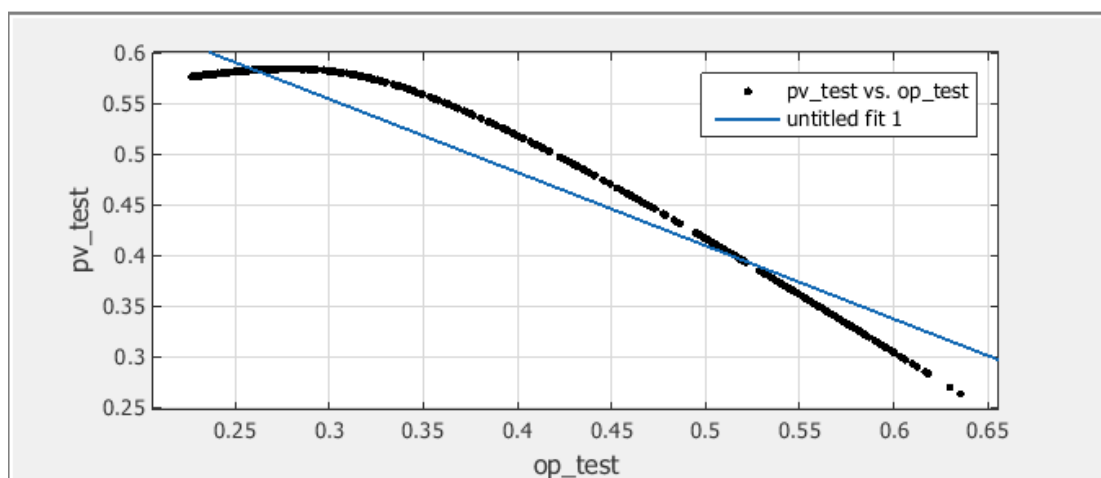


Figure 4.15 Polynomial plot for loop 4

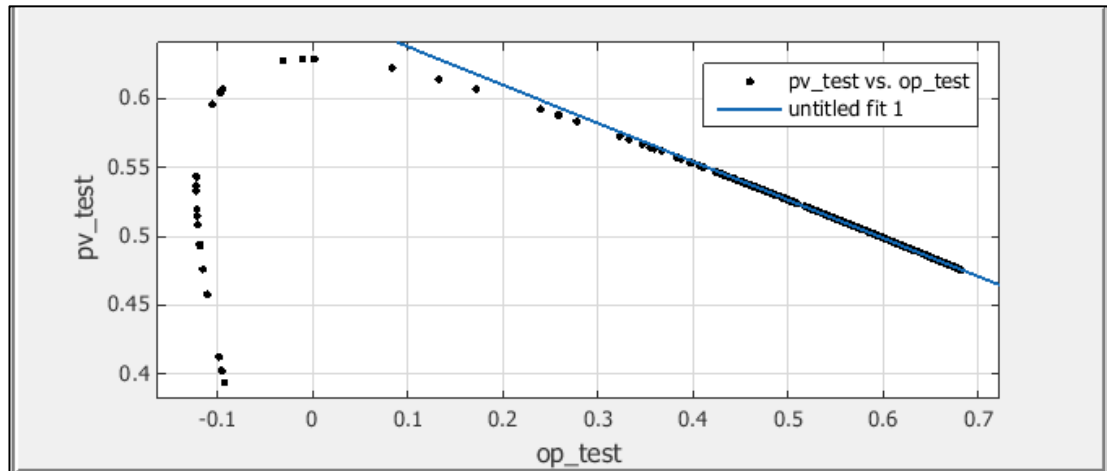


Figure 4.16 Polynomial plot for loop 5

As what have been described earlier in chapter 3, loop 1 is the system of non-stiction followed by loop 2 and loop 3 for stiction system. Then, loop 4 and loop 5 are system with external disturbances. Table 4.6 shows the results of the errors calculation for each case.

Table 4.6 Error calculation for each industrial case study

Error Calculation	Industrial Loop 1	Industrial Loop 2	Industrial Loop 3	Industrial Loop 4	Industrial Loop 5
<b>R2</b>	1	0.03121	1	0.9316	0.3684
<b>SSE</b>	5.41E-06	18.8	1.44E-07	3.30E-01	0.2827
<b>RMSE</b>	7.33E-05	0.15	3.22E-05	2.57E-02	0.02371

Based on the data above, it can be seen that for regression values of loop 1, loop 3 and loop 4 are approaching 1. Meanwhile, for loop 2 and loop5, the values of  $R^2$  are very small. However, this is violating the observations made earlier as for non-stiction system, the  $R^2$  values should be approaching 1 and stiction system, in this case is loop 3 should not have the value of 1. This is maybe due to the limitations of NLPCA to generate the pv and op data as the number of samples for this system is very small which is 200 samples compared to other system. This is the main drawback for NLPCA, thus resulting in the incorrect indications for the real industrial data. Hence, the  $R^2$  calculation is not suitable to detect stiction system.

Another error calculation is based on the SSE. Previously, the stiction system has the highest values among others while for non-stiction system, the SSE is very small for tightly-tuned controller and slightly higher for presence of disturbances. However, again this observation cannot be applied here as loop 4 also has the SSE values almost similar to loop 1. Hence, the SSE calculations cannot be the main reference to detect any stiction in the system.

Lastly is the RMSE calculation. For the simulation case study, for any values that is more than 0.1 is considered as stiction system. Here, we can see that for loop2, it has values higher than 0.1 which is 0.15. Meanwhile, for loop 3, there is still some drawback due to the small number of data of the system. The RMSE for both external disturbances cases have lower values than 0.1. This had eventually proven the efficiency of RMSE to detect stiction system in the real industrial data.

In a nutshell,  $R^2$  and SSE play an important role in supporting the detection techniques. However, RMSE is the most essential calculation to be done for detection method due to the capability of it in detecting stiction system especially to differentiate with other presence of disturbances and tightly-tuned controller conditions.

The proposed error testing method is further compared with existing detection method which is the fuzzy clustering method mentioned earlier in Table 2.2 in chapter 2. Table 4.7 is the comparison table whereby the red colour indicates incorrect result and green colour shows the correct result.

Table 4.7 Comparison of the proposed method with fuzzy clustering method on the industrial data set

Loop number	Fuzzy Clustering Method	Error Testing Method
Loop 1	NO	NO
Loop 2	YES	YES
Loop 3	YES	NO
Loop 4	NO	NO
Loop 5	NO	NO

As been stated earlier, the proposed method has some limitations for a small set of data. However, for other loops, it manages to generate correct output. This has strongly supports the effectiveness of this method as the new stiction detection method using statistical-based method especially on the real industrial data.

## **CHAPTER 5**

### **CONCLUSIONS AND RECOMMENDATION**

#### **5.1 Conclusion**

As a conclusion, there are two statistical-based methods which have been used for fault detection and now have been applied to detect control valve stiction. However, for GLR testing method, it is not suitable to detect stiction system due to the constraint of detecting the presence of oscillatory disturbances as stiction system too. Meanwhile, for Error Testing Method with additional slope calculation, this method managed to differentiate each simulation and real industrial case studies for stiction or non-stiction system. It has proven to be a reliable tool for stiction detection. Hence, based on the objectives stated earlier, it can be said that this project had achieved the objectives mentioned earlier.

#### **5.2 Recommendation**

One of the disadvantages of NLPCA testing method is due to the limitations of generating based on smaller number of data. This drawback has effect the data generated whereby the op and pv data does not fully portray the conditions of the system. Hence, it is recommended to test other statistical-based methods with other model in generating the data such as shape-based method and Hammerstein-based model.

Since there are some limitations and time constraints to expand this project, it is recommended for the next research to continue on this application for modelling and compensation of stiction.

## REFERENCES

- [1] M. A. Paulonis and J. W. Cox, "A practical approach for large-scale controller performance assessment, diagnosis, and improvement," *Journal of Process Control*, vol. 13, pp. 155-168, 3/ 2003.
- [2] N. F. T. S. L. S. M. A. A. Shoukat Choudhury, "Modelling Valve Stiction," *Publication in Control Engineering Practice*, 2004.
- [3] J. F. Smuts, "Troubleshooting and Solving Poor Control Loop Performance," *Instrumentation & Controls*, 06/01/2013 2013.
- [4] H. Zabiri and M. Ramasamy, "NLPCA as a diagnostic tool for control valve stiction," *Journal of Process Control*, vol. 19, pp. 1368-1376, 9/ 2009.
- [5] F. Controls, "Control Valve Handbook Fourth Edition [Z]," *Iowa, United States: Emerson Process Management*, 2005.
- [6] S. A. A. Choudhury, S. Shah, and N. Thornhill, "Different Types of Faults in Control Valves," in *Diagnosis of Process Nonlinearities and Valve Stiction*, ed: Springer Berlin Heidelberg, 2008, pp. 137-141.
- [7] M. D. Holloway, C. Nwaoha, and O. A. Onyewuenyi, *Process plant equipment: operation, control, and reliability*: John Wiley & Sons, 2012.
- [8] J. M. D. Katzman, J. Noelanders & M. W. Galant, "Valve Types Selection," *Michigan Chemical Process Dynamics and Controls*, 05/09/2007 2007.
- [9] T. B. R. S. Patwardhan, F. Al-Anazi & T. J. Schroeder, "Managing the Performance of Control Loops with Valve Stiction: An Industrial Perspective," 2014.
- [10] S. J. Qin, "Control performance monitoring—a review and assessment," *Computers & Chemical Engineering*, vol. 23, pp. 173-186, 1998.
- [11] S. M. A. A. a. Choudhury, S. L. a. Shah, and N. F. a. Thornhill, *Diagnosis of Process Nonlinearities and Valve Stiction Data Driven Approaches*, 2008.
- [12] A. Singhal and T. I. Salsbury, "A simple method for detecting valve stiction in oscillating control loops," *Journal of Process Control*, vol. 15, pp. 371-382, 2005.
- [13] M. Jelali, "Estimation of valve stiction in control loops using separable least-squares and global search algorithms," *Journal of Process Control*, vol. 18, pp. 632-642, 8/2008.

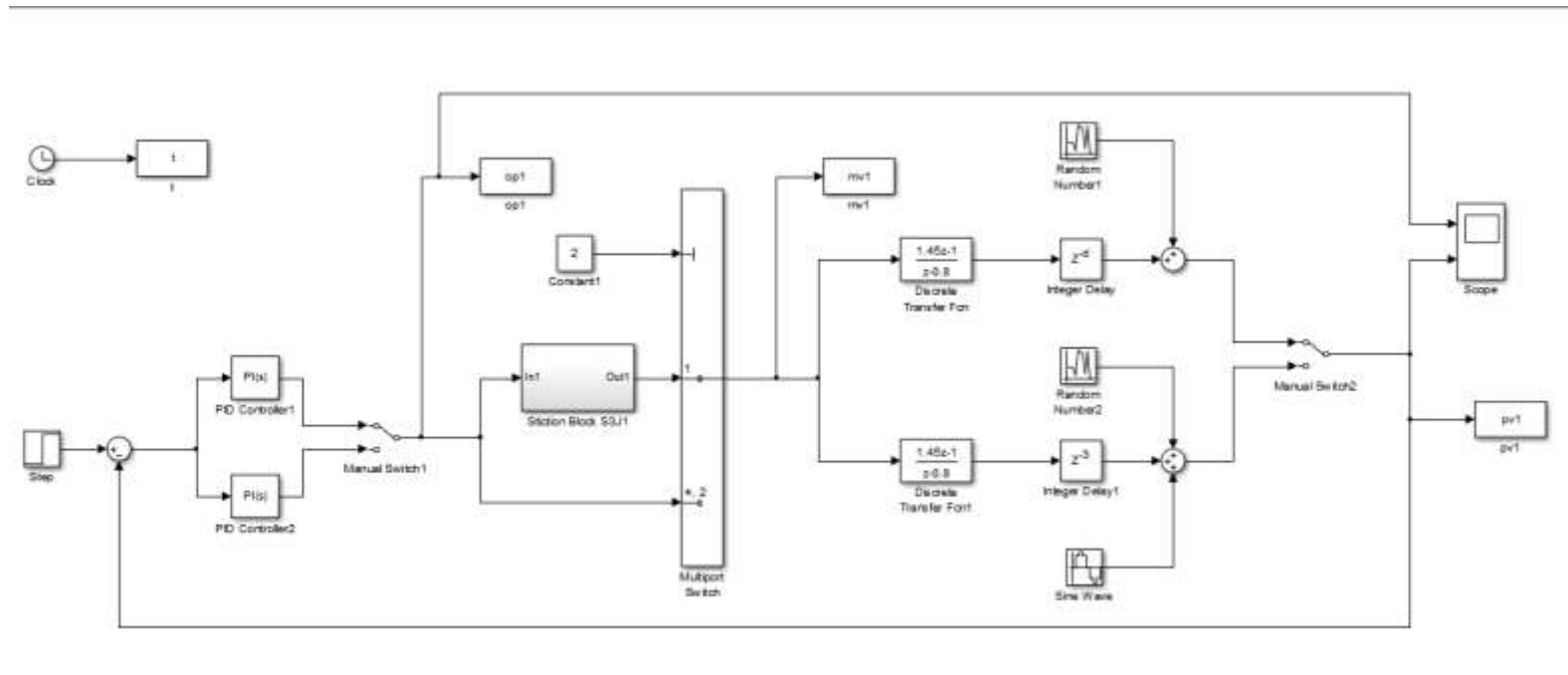
- [14] M. A. Daneshwar and N. Mohd Noh, "Detection of stiction in flow control loops based on fuzzy clustering," *Control Engineering Practice*, vol. 39, pp. 23-34, 6/2015.
- [15] A. Horch, "A simple method for detection of stiction in control valves," *Control Engineering Practice*, vol. 7, pp. 1221-1231, 10/1999.
- [16] B. Srinivasan, U. Nallasivam, and R. Rengaswamy, "Multiple root cause analysis of linear oscillatory closed-loop single-input single-output (SISO) systems," in *Advanced Control of Industrial Processes (ADCONIP), 2011 International Symposium on*, 2011, pp. 36-41.
- [17] S. Kalaivani, T. Aravind, and D. Yuvaraj, "A Single Curve Piecewise Fitting Method for Detecting Valve Stiction and Quantification in Oscillating Control Loops," in *Proceedings of the Second International Conference on Soft Computing for Problem Solving (SocProS 2012), December 28-30, 2012*. vol. 236, B. V. Babu, A. Nagar, K. Deep, M. Pant, J. C. Bansal, K. Ray, *et al.*, Eds., ed: Springer India, 2014, pp. 13-24.
- [18] M. Bartyś, J. Quevedo, and R. Patton, "Development and application of methods for actuator diagnosis in industrial control systems (DAMADICS): A benchmark study," *SafeProcess 2003, Washington DC*, 2003.
- [19] R. Isermann, "Model-based fault-detection and diagnosis—status and applications," *Annual Reviews in control*, vol. 29, pp. 71-85, 2005.
- [20] S. Katipamula and M. R. Brambley, "Review article: methods for fault detection, diagnostics, and prognostics for building systems—a review, Part I," *HVAC&R Research*, vol. 11, pp. 3-25, 2005.
- [21] L. Bo-Ang and Y. Hao, "Statistical  $\chi^2$  Testing Based Fault Detection for Linear Discrete Time-delay Systems," *Acta Automatica Sinica*, vol. 40, pp. 1278-1284, 2014.
- [22] H. J. Galicia, Q. P. He, and J. Wang, "A comprehensive evaluation of Statistics Pattern Analysis based process monitoring," in *International Symposium on Advanced Control of Chemical Processes ADCHEM*, 2012.
- [23] F. Harrou, M. N. Nounou, H. N. Nounou, and M. Madakyaru, "Statistical fault detection using PCA-based GLR hypothesis testing," *Journal of Loss Prevention in the Process Industries*, vol. 26, pp. 129-139, 2013.
- [24] C. Svärd, M. Nyberg, E. Frisk, and M. Krysander, "Data-driven and adaptive statistical residual evaluation for fault detection with an automotive



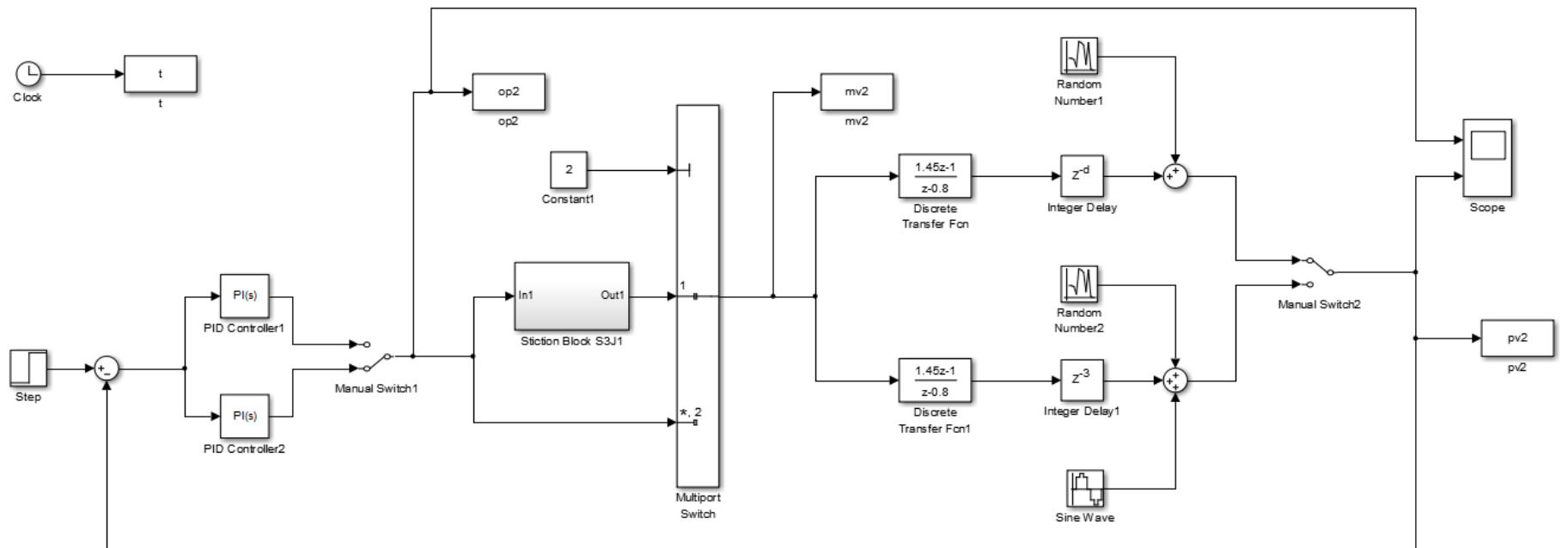
- application," *Mechanical systems and signal processing*, vol. 45, pp. 170-192, 2014.
- [25] Y. Zhao, S. Wang, and F. Xiao, "A statistical fault detection and diagnosis method for centrifugal chillers based on exponentially-weighted moving average control charts and support vector regression," *Applied Thermal Engineering*, vol. 51, pp. 560-572, 3/2013.
- [26] A. K. A. P. Singh, "Neural Network based Fault Diagnosis in Analog Electronic Circuit using Polynomial Curve Fitting," *International Journal of Computer Applications*, vol. 61, 2013.
- [27] M. M. Li and B. Verma, "Nonlinear curve fitting to stopping power data using RBF neural networks," *Expert Systems with Applications*, vol. 45, pp. 161-171, 3/1/2016.

## APPENDICES

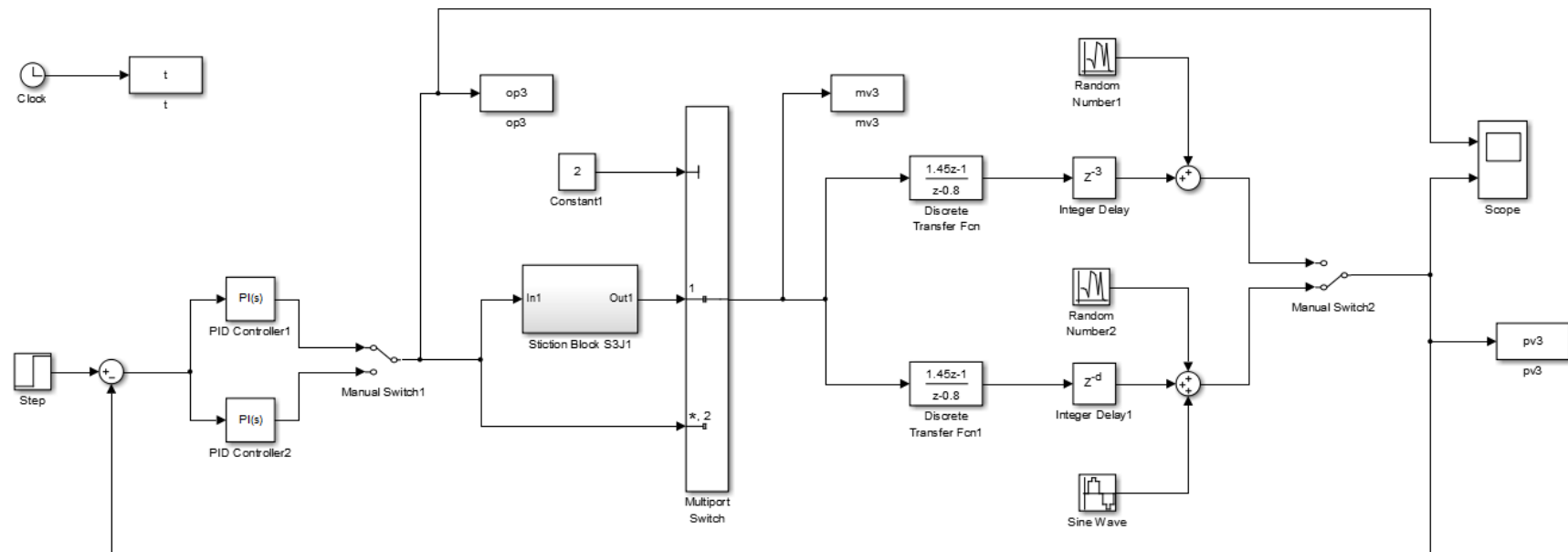
### APPENDIX A



Block Diagram for Base Case



Block Diagram for Case 1



Block Diagram for Case 2



## APPENDIX B

```
close all; clear; clc;
```

```
Ts = 1;
```

```
%%Simulink model
% sim('system_4_case3');
% sim('system_3_case2');
% sim('system_2_case1');
% sim('system_1_basecase');
```

```
%%Fuzzy Clustering Data
% A = xlsread('loop1_noS');
% A =
xlsread('loop2_stiction');
A = xlsread('loop3_stiction');
% A =
xlsread('loop4_stiction');
% A = xlsread('loop5_dist');
% A = xlsread('loop6_dist');
```

```
[Arow,Acol]=size(A);
u1 = A(1:Arow,2);
y1 = A(1:Arow,1);
```

```
% Extracting data
op = u1;
pv = y1;
```

```
% Normalization
min_pv = min(pv); min_op =
min(op);
max_pv = max(pv); max_op =
max(op);
```

```
[row1,col1] = size(op);
norm_op = zeros(row1,col1);
i = 1;
for i=1:row1
    norm_op(i,:) = (op(i,:) -
min_op)/(max_op - min_op);
    i = i+1;
end
```

```
[row2,col2] = size(pv);
norm_pv = zeros(row2,col2);
j = 1;
for j=1:row2
    norm_pv(j,:) = (pv(j,:) -
min_pv)/(max_pv - min_pv);
    j = j+1;
end
```

```
% creating index no of original
sequence
indexno = ones(row1,1);
for l=2:row1;
```

```
indexno(l,:) = indexno(l-
1,:)+1;
end
```

```
% create a random number vector
with the same size as op
% use rand = between 0 to 1
randno = rand(row1,1);
```

```
% create intermediate data
matrix for randomization
datamat = [randno indexno
norm_op norm_pv];
```

```
% sort data according to random
numbers
```

```
A = sortrows(datamat,1);
A = datamat;
```

```
%
% creating the data matrix
[row,col] = size(A);
L_tr = round(0.7*row);
```

```
% extracting T and V sets
T_set = A(1:L_tr,2:4);
V_set = A(L_tr+1:row,2:4);
```

```
% re-arrange the original time
sequence
```

```
T_setr = T_set;
V_setr = V_set;
% T_setr = sortrows(T_set,1);
% V_setr = sortrows(V_set,1);
```

```
% NLPCA
```

```
P_tr = T_setr(:,2:3)';
T_tr = T_setr(:,2:3)';
P_v = V_setr(:,2:3)';
T_v = V_setr(:,2:3)';
```

```
% naming TF
```

```
p1 = 'purelin'; t = 'tansig';
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%% Setup network
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%% NLPCA
```

```
%Figure (1)
```

```
net=newff(minmax(P_tr), [2 3 1
3
2],{t,t,p1,t,p1},'trainrp','lea
rngdm','mse'); % 5 layers
% net = init(net);
% net.initFcn='initlay';
%
net.layers{1}.initFcn='initnw';
% net = init(net);
% % net = initnw(net,1);
```

```

% rand('seed',419877);
rng('default');
rng(1,'twister');
net = init(net);
net.trainParam.show=5;
net.trainParam.epochs=1000;
net.trainParam.goal=1e-4;
val.P = P_v;
val.T = T_v;
[net tr] = train(
net,P_tr,T_tr,[],[],val);
a = sim(net,P_tr);
figure(1)
xlabel('op');
ylabel('pv');
subplot(5,2,1),scatter(a(1,:),a
(2,:),5,'r','filled');
drawnow; hold on;
clear('net');

net1=newff(minmax(P_tr), [2 3 1
3
2],{t,t,p1,t,p1},'trainrp','lea
rngdm','mse'); % 5 layers
% rand('seed',419877);
% rng('default');
rng(2,'twister');
net1 = init(net1);
net1.trainParam.show=5;
net1.trainParam.epochs=1000;
net1.trainParam.goal=1e-4;
val.P = P_v;
val.T = T_v;
[net1 tr] = train(
net1,P_tr,T_tr,[],[],val);
a1 = sim(net1,P_tr);
figure(1)
xlabel('op');
ylabel('pv');
subplot(5,2,2),scatter(a1(1,:),
a1(2,:),5,'r','filled');
drawnow; hold on;
clear('net1');

net2=newff(minmax(P_tr), [2 3 1
3
2],{t,t,p1,t,p1},'trainrp','lea
rngdm','mse'); % 5 layers
% rand('seed',419877);
% rng('default');
rng(3,'twister');
net2 = init(net2);
net2.trainParam.show=5;
net2.trainParam.epochs=1000;
net2.trainParam.goal=1e-4;
val.P = P_v;
val.T = T_v;
[net2 tr] = train(
net2,P_tr,T_tr,[],[],val);
a2 = sim(net2,P_tr);

figure(1)
xlabel('op');
ylabel('pv');
subplot(5,2,3),scatter(a2(1,:),
a2(2,:),5,'r','filled');
drawnow; hold on;
clear('net2');

net3=newff(minmax(P_tr), [2 3 1
3
2],{t,t,p1,t,p1},'trainrp','lea
rngdm','mse'); % 5 layers
% rand('seed',419877);
% rng('default');
rng(4,'twister');
net3 = init(net3);
net3.trainParam.show=5;
net3.trainParam.epochs=1000;
net3.trainParam.goal=1e-4;
val.P = P_v;
val.T = T_v;
[net3 tr] = train(
net3,P_tr,T_tr,[],[],val);
a3 = sim(net3,P_tr);
figure(1)
xlabel('op');
ylabel('pv');
subplot(5,2,4),scatter(a3(1,:),
a3(2,:),5,'r','filled');
drawnow; hold on;
clear('net3');

net4=newff(minmax(P_tr), [2 3 1
3
2],{t,t,p1,t,p1},'trainrp','lea
rngdm','mse'); % 5 layers
% rand('seed',419877);
% rng('default');
rng(5,'twister');
net4 = init(net4);
net4.trainParam.show=5;
net4.trainParam.epochs=1000;
net4.trainParam.goal=1e-4;
val.P = P_v;
val.T = T_v;
[net4 tr] = train(
net4,P_tr,T_tr,[],[],val);
a4 = sim(net4,P_tr);
figure(1)
xlabel('op');
ylabel('pv');
subplot(5,2,5),scatter(a4(1,:),
a4(2,:),5,'r','filled');
drawnow; hold off;
clear('net4');

net5=newff(minmax(P_tr), [2 3 1
3
2],{t,t,p1,t,p1},'trainrp','lea
rngdm','mse'); % 5 layers

```

```

% rand('seed',419877);
% rng('default');
rng(6, 'twister');
net5 = init(net5);
net5.trainParam.show=5;
net5.trainParam.epochs=1000;
net5.trainparam.goal=1e-4;
xlabel('op');
ylabel('pv');
[net5 tr] = train(
net5,P_tr,T_tr,[],[],val);
a5 = sim(net5,P_tr);
figure(1)
xlabel('u');
ylabel('y');
subplot(5,2,6),scatter(a5(1,:),
a5(2,:),5,'r','filled');
drawnow; hold off;
clear('net5');

net6=newff(minmax(P_tr), [2 3 1
3
2],{t,t,p1,t,p1},'trainrp','lea
rngdm','mse'); % 5 layers
% rand('seed',419877);
% rng('default');
rng(7, 'twister');
net6 = init(net6);
net6.trainParam.show=5;
net6.trainParam.epochs=1000;
net6.trainparam.goal=1e-4;
val.P = P_v;
val.T = T_v;
[net6 tr] = train(
net6,P_tr,T_tr,[],[],val);
a6 = sim(net6,P_tr);
figure(1)
xlabel('op');
ylabel('pv');
subplot(5,2,7),scatter(a6(1,:),
a6(2,:),5,'r','filled');
drawnow; hold off;
clear('net6');

net7=newff(minmax(P_tr), [2 3 1
3
2],{t,t,p1,t,p1},'trainrp','lea
rngdm','mse'); % 5 layers
% rand('seed',419877);
% rng('default');
rng(8, 'twister');
net7 = init(net7);
net7.trainParam.show=5;
net7.trainParam.epochs=1000;
net7.trainparam.goal=1e-4;
val.P = P_v;
val.T = T_v;
[net7 tr] = train(
net7,P_tr,T_tr,[],[],val);
a7 = sim(net7,P_tr);

figure(1)
xlabel('op');
ylabel('pv');
subplot(5,2,8),scatter(a7(1,:),
a7(2,:),5,'r','filled');
drawnow; hold off;
clear('net7');

net8=newff(minmax(P_tr), [2 3 1
3
2],{t,t,p1,t,p1},'trainrp','lea
rngdm','mse'); % 5 layers
% rand('seed',419877);
% rng('default');
rng(9, 'twister');
net8 = init(net8);
net8.trainParam.show=5;
net8.trainParam.epochs=1000;
net8.trainparam.goal=1e-4;
val.P = P_v;
val.T = T_v;
[net8 tr] = train(
net8,P_tr,T_tr,[],[],val);
a8 = sim(net8,P_tr);
figure(1)
xlabel('op');
ylabel('pv');
subplot(5,2,9),scatter(a8(1,:),
a8(2,:),5,'r','filled');
drawnow; hold off;
clear('net8');

net9=newff(minmax(P_tr), [2 3 1
3
2],{t,t,p1,t,p1},'trainrp','lea
rngdm','mse'); % 5 layers
% rand('seed',419877);
% rng('default');
rng(10, 'twister');
net9 = init(net9);
net9.trainParam.show=5;
net9.trainParam.epochs=1000;
net9.trainparam.goal=1e-4;
val.P = P_v;
val.T = T_v;
[net9 tr] = train(
net9,P_tr,T_tr,[],[],val);
a9 = sim(net9,P_tr);
figure(1)
xlabel('op');
ylabel('pv');
subplot(5,2,10),scatter(a9(1,:),
a9(2,:),5,'r','filled');
drawnow; hold off;
clear('net9');

% Method 1 - Getting the best
fit
% Calculating the value of SSE
and Residual for deg 1

```



```
% Get the yfit and R^2 for each
data from a-a9
```

```
x = a(1,:);
y = a(2,:);
p1 = polyfit (x,y,1);
yfit1 = polyval (p1,x);
% yfit = p1(1)*x + p1(2);
yresid1 = y - yfit1;
SSresid_a1 = sum(yresid1.^2);
SStotal_a1 = (length(y)-
1)*var(y);
rsq_a1 = 1 -
SSresid_a1/SStotal_a1;
disp (rsq_a1);
pla = polyfit (x,y,2);
yfit1a = polyval (pla,x);
yresid1a = y - yfit1a;
SSresid_a1a = sum(yresid1a.^2);
SStotal_a1a = (length(y)-
1)*var(y);
rsq_a1a = 1 -
SSresid_a1a/SStotal_a1a;
yfit1b = p1(2);
yresid1b = y - yfit1b;
SSresid_a1b = sum(yresid1b.^2);
SStotal_a1b = (length(y)-
1)*var(y);
rsq_a1b = 1 -
SSresid_a1b/SStotal_a1b;

x1 = a1(1,:);
y1 = a1(2,:);
p2 = polyfit (x1,y1,1);
yfit2 = polyval (p2,x1);
% yfit1 = p1(1)*x1 + p1(2);
yresid2 = y1 - yfit2;
SSresid_a2 = sum(yresid2.^2);
SStotal_a2 = (length(y1)-
1)*var(y1);
rsq_a2 = 1 -
SSresid_a2/SStotal_a2;
disp (rsq_a2);
p2a = polyfit (x1,y1,2);
yfit2a = polyval (p2a,x1);
yresid2a = y1 - yfit2a;
SSresid_a2a = sum(yresid2a.^2);
SStotal_a2a = (length(y1)-
1)*var(y1);
rsq_a2a = 1 -
SSresid_a2a/SStotal_a2a;

x2 = a2(1,:);
y2 = a2(2,:);
p3 = polyfit (x2,y2,1);
yfit3 = polyval (p3,x2);
% yfit2 = p2(1)*x2 + p2(2);
yresid3 = y2 - yfit3;
SSresid_a3 = sum(yresid3.^2);
```

```
SStotal_a3 = (length(y2)-
1)*var(y2);
rsq_a3 = 1 -
SSresid_a3/SStotal_a3;
disp (rsq_a3);
p3a = polyfit (x2,y2,2);
yfit3a = polyval (p3a,x2);
yresid3a = y2 - yfit3a;
SSresid_a3a = sum(yresid3a.^2);
SStotal_a3a = (length(y2)-
1)*var(y2);
rsq_a3a = 1 -
SSresid_a3a/SStotal_a3a;
```

```
x3 = a3(1,:);
y3 = a3(2,:);
p4 = polyfit (x3,y3,1);
yfit4 = polyval (p4,x3);
% yfit3 = p3(1)*x3 + p3(2);
yresid4 = y3 - yfit4;
SSresid_a4 = sum(yresid4.^2);
SStotal_a4 = (length(y3)-
1)*var(y3);
rsq_a4 = 1 -
SSresid_a4/SStotal_a4;
disp (rsq_a4);
p4a = polyfit (x3,y3,2);
yfit4a = polyval (p4a,x3);
yresid4a = y3 - yfit4a;
SSresid_a4a = sum(yresid4a.^2);
SStotal_a4a = (length(y3)-
1)*var(y3);
rsq_a4a = 1 -
SSresid_a4a/SStotal_a4a;
```

```
x4 = a4(1,:);
y4 = a4(2,:);
p5 = polyfit (x4,y4,1);
yfit5 = polyval (p5,x4);
% yfit4 = p4(1)*x4 + p4(2);
yresid5 = y4 - yfit5;
SSresid_a5 = sum(yresid5.^2);
SStotal_a5 = (length(y4)-
1)*var(y4);
rsq_a5 = 1 -
SSresid_a5/SStotal_a5;
disp (rsq_a5);
p5a = polyfit (x4,y4,2);
yfit5a = polyval (p5a,x4);
yresid5a = y4 - yfit5a;
SSresid_a5a = sum(yresid5a.^2);
SStotal_a5a = (length(y4)-
1)*var(y4);
rsq_a5a = 1 -
SSresid_a5a/SStotal_a5a;
```

```
x5 = a5(1,:);
y5 = a5(2,:);
p6 = polyfit (x5,y5,1);
yfit6 = polyval (p6,x5);
```

```

% yfit5 = p5(1)*x5 + p5(2);
yresid6 = y5 - yfit6;
SSresid_a6 = sum(yresid6.^2);
SStotal_a6 = (length(y5)-
1)*var(y5);
rsq_a6 = 1 -
SSresid_a6/SStotal_a6;
disp (rsq_a6);
p6a = polyfit (x5,y5,2);
yfit6a = polyval (p5a,x5);
yresid6a = y5 - yfit6a;
SSresid_a6a = sum(yresid6a.^2);
SStotal_a6a = (length(y5)-
1)*var(y5);
rsq_a6a = 1 -
SSresid_a6a/SStotal_a6a;

x6 = a6(1,:);
y6 = a6(2,:);
p7 = polyfit (x6,y6,1);
yfit7 = polyval (p7,x6);
% yfit6 = p6(1)*x6 + p6(2);
yresid7 = y6 - yfit7;
SSresid_a7 = sum(yresid7.^2);
SStotal_a7 = (length(y6)-
1)*var(y6);
rsq_a7 = 1 -
SSresid_a7/SStotal_a7;
disp (rsq_a7);
p7a = polyfit (x6,y6,2);
yfit7a = polyval (p7a,x6);
yresid7a = y6 - yfit7a;
SSresid_a7a = sum(yresid7a.^2);
SStotal_a7a = (length(y6)-
1)*var(y6);
rsq_a7a = 1 -
SSresid_a7a/SStotal_a7a;

x7 = a7(1,:);
y7 = a7(2,:);
p8 = polyfit (x7,y7,1);
yfit8 = polyval (p8,x7);
% yfit7 = p7(1)*x7 + p7(2);
yresid8 = y7 - yfit8;
SSresid_a8 = sum(yresid8.^2);
SStotal_a8 = (length(y7)-
1)*var(y7);
rsq_a8 = 1 -
SSresid_a8/SStotal_a8;
disp (rsq_a8);
p8a = polyfit (x7,y7,2);
yfit8a = polyval (p8a,x7);
yresid8a = y7 - yfit8a;
SSresid_a8a = sum(yresid8a.^2);
SStotal_a8a = (length(y7)-
1)*var(y7);
rsq_a8a = 1 -
SSresid_a8a/SStotal_a8a;

x8 = a8(1,:);
y8 = a8(2,:);
p9 = polyfit (x8,y8,1);
yfit9 = polyval (p9,x8);
% yfit8 = p8(1)*x8 + p8(2);
yresid9 = y8 - yfit9;
SSresid_a9 = sum(yresid9.^2);
SStotal_a9 = (length(y8)-
1)*var(y8);
rsq_a9 = 1 -
SSresid_a9/SStotal_a9;
disp (rsq_a9);
p9a = polyfit (x8,y8,2);
yfit9a = polyval (p9a,x8);
yresid9a = y8 - yfit9a;
SSresid_a9a = sum(yresid9a.^2);
SStotal_a9a = (length(y8)-
1)*var(y8);
rsq_a9a = 1 -
SSresid_a9a/SStotal_a9a;

x9 = a9(1,:);
y9 = a9(2,:);
p10 = polyfit (x9,y9,1);
yfit10 = polyval (p10,x);
% yfit9 = p9(1)*x9 + p9(2);
yresid10 = y9 - yfit10;
SSresid_a10 = sum(yresid10.^2);
SStotal_a10 = (length(y9)-
1)*var(y9);
rsq_a10 = 1 -
SSresid_a10/SStotal_a10;
disp (rsq_a10);
p10a = polyfit (x9,y9,2);
yfit10a = polyval (p10a,x);
yresid10a = y9 - yfit10a;
SSresid_a10a =
sum(yresid10a.^2);
SStotal_a10a = (length(y9)-
1)*var(y9);
rsq_a10a = 1 -
SSresid_a10a/SStotal_a10a;
yfit10b = p10(2);
yresid10b = y9 - yfit10b;
SSresid_a10b =
sum(yresid10b.^2);
SStotal_a10b = (length(y9)-
1)*var(y9);
rsq_a10b = 1 -
SSresid_a10b/SStotal_a10b;

% Identify min and max values

data_loop1pulp =
[rsq_a1,rsq_a2,rsq_a3,rsq_a4,rs
q_a5,rsq_a6,rsq_a7,rsq_a8,rsq_a
9,rsq_a10];
data_loop1pulpa =
[rsq_a1a,rsq_a2a,rsq_a3a,rsq_a4
a,rsq_a5a,rsq_a6a,rsq_a7a,rsq_a
8a,rsq_a9a,rsq_a10a];

```

```

data_loop1pulp_sse =
[SSresid_a1,SSresid_a2,SSresid_
a3,SSresid_a4,SSresid_a5,SSresi
d_a6,SSresid_a7,SSresid_a8,SSre
sid_a9,SSresid_a10];
data_loop1pulpa_sse =
[SSresid_a1a,SSresid_a2a,SSresi
d_a3a,SSresid_a4a,SSresid_a5a,S
Sresid_a6a,SSresid_a7a,SSresid_
a8a,SSresid_a9a,SSresid_a10a];
data_des = sort(data_loop1pulp,
'descend');
a_all =
[a;a1;a2;a3;a4;a5;a6;a7;a8;a9];

[minR indmin] =
min(data_loop1pulp);
[maxR indmax] =
max(data_loop1pulp);

% Choosing R2min

if minR == rsq_a1;
    R2min = rsq_a1;
elseif minR == rsq_a2;
    R2min = rsq_a2;
elseif minR == rsq_a3;
    R2min = rsq_a3;
elseif minR == rsq_a4;
    R2min = rsq_a4;
elseif minR == rsq_a5;
    R2min = rsq_a5;
elseif minR == rsq_a6;
    R2min = rsq_a6;
elseif minR == rsq_a7;
    R2min = rsq_a7;
elseif minR == rsq_a8;
    R2min = rsq_a8;
elseif minR == rsq_a9;
    R2min = rsq_a9;
else
    R2min = rsq_a10;
end

% Calculating R2<0.1
iir2 = 1;
iisse = 1;
for ir2 =
1:length(data_loop1pulp)
    if data_loop1pulp(ir2)<0.5;
        ir2ind(iir2)=1;
        iir2=iir2+1;
    end
    if
(data_loop1pulp_sse(ir2)-
data_loop1pulpa_sse(ir2))>0;
        isseind(iisse)=1;
        iisse=iisse+1;
    end
end

% Testing the condition for
curve fitting
%
% if ((mean(data_loop1pulp)-
mean(data_loop1pulpa))<0) &&
((mean(data_loop1pulp_sse)-
mean(data_loop1pulpa_sse))>0);
%     display('Stiction may
exist')
% %     break;
%
% else
%     display('No stiction')
%
% end

% if sum(ir2ind) >= 5;
%     display('No stiction')
% %     break;
%
% else
%     display('Stiction may
exist')
%
% end

% if (sum(ir2ind)>=1) &&
(sum(iisseind)>=1);
%     display('Stiction may
exist')
% %     break;
%
% else
%     display('No stiction')
%
% end

%
if minR >= 0.5;
    display('No stiction')
%     break;
%
% else
%     R2min = minR;
%     disp (R2min);
end

% set variables for cftool
datamin = a_all(indmin*2-
1:indmin*2,:);
pv_test = datamin(2,:);
op_test = datamin(1,:);

cftool

figure(2)
plot (op_test, pv_test);

```